实用第一 智慧密集 Practicability First, Intelligence Intensive

# 电脑编程技巧与维护

COMPUTER PROGRAMMING SKILLS & MAINTENANCE

http://www.comprg.com.cn

上 8月 2013年8月03日

每期定价:11.00元 全年定价:264.00元
《电脑编程技巧与维护》杂志社出版
刊号: ISSN 1006-4052 CN11-3411/TP
广告许可证 京海工商广字0151

国家级科技期刊 中国学术期刊综合评价数据库统计源期刊 中国核心期刊（遴选）数据库收录期刊



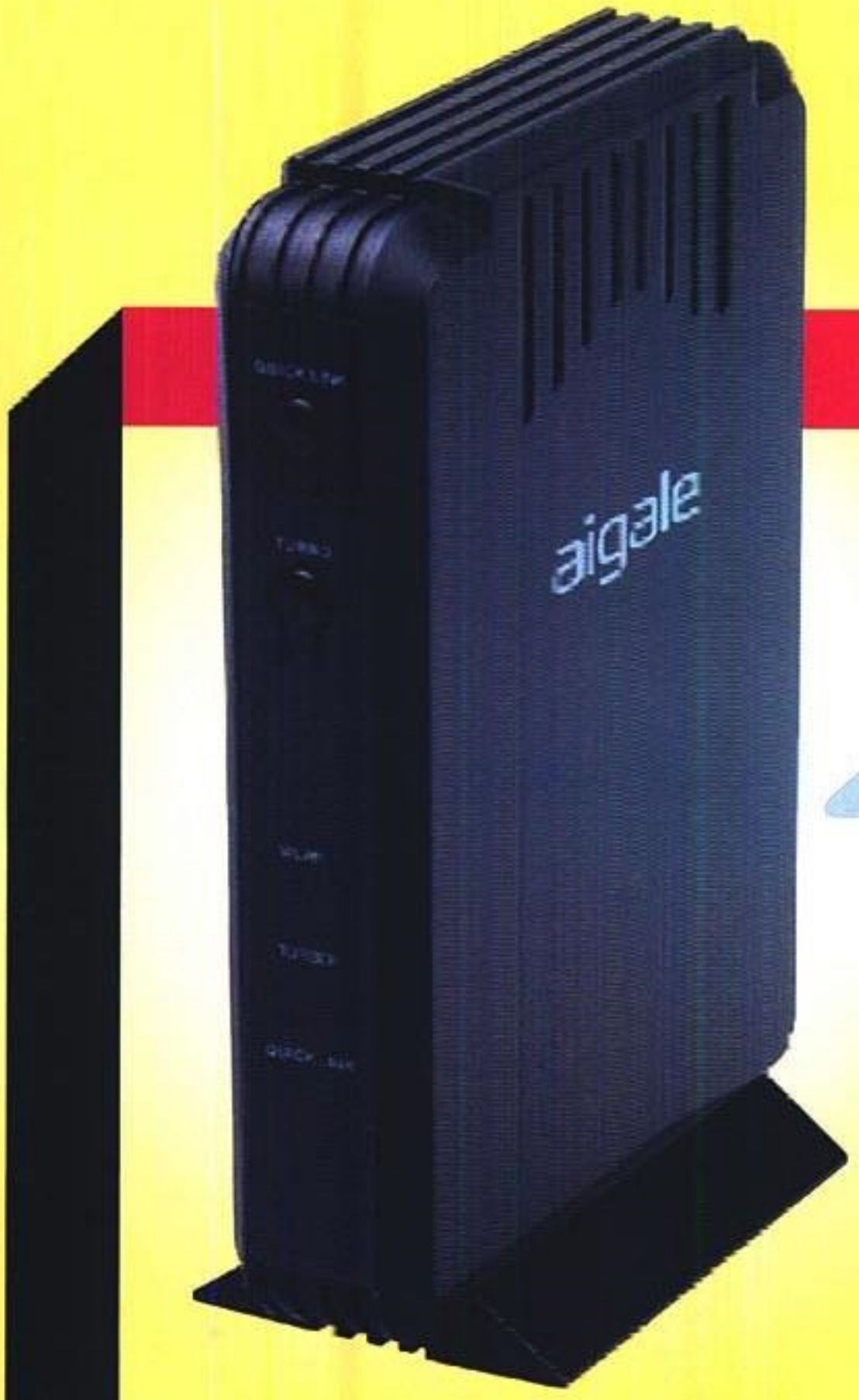





ISSN 1006-4052

# 抢先Hold住PCWorld

## 即可精巧“联”通科技未来




**现在邮购2013年**
《微电脑世界》**全年杂志**

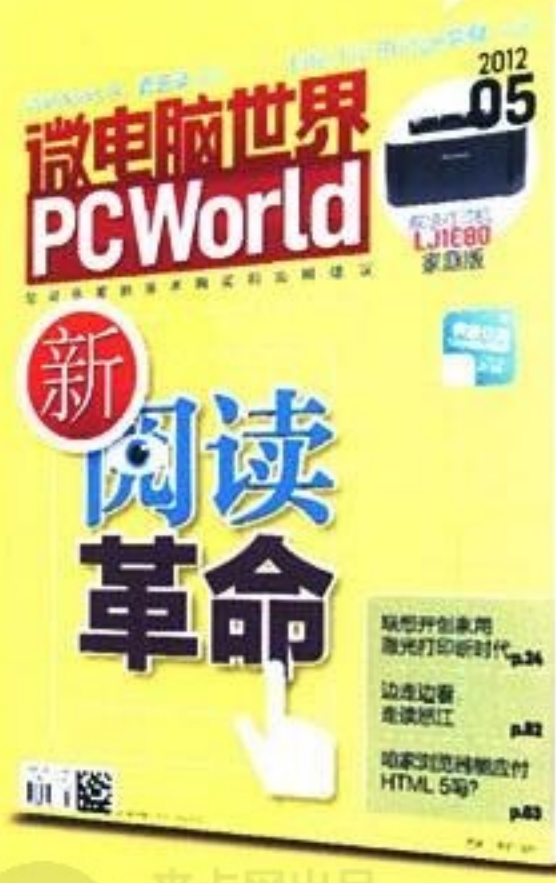
**即得一个价值149元**

**海联达Ai-R100 极风**
# 无线路由器

轻松联通您的智能终端，
让您尊享全球IT资深顾问
随时随地的贴身资讯服务。

飞信 Fetion
**用户可登陆：**
http://sms.feixin.10086.cn/Subscribe/getInfo/id/215 订阅
中国移动用户发送ZZ到125200915，即可订阅

**汇款地址：**北京市123信箱，收款人：微电脑世界杂志，邮编：100036
**杂志定价：**144元/年（12元/月）
**活动咨询：**周一到周五，9：00~11：30，13：00~17：30
**电话：**010-88230131
**杂志社现场订阅地址：**北京市海淀区万寿路翠微中里14号楼
**在线订阅：**http://www.pcworld.com.cn/about/ebuy/pay.html




**活动说明：**

**活动时间：**2012年6月10日~2013年6月30日（邮局汇款以邮戳为准）
在汇款单附言栏注明“2013年微电脑世界”，同时留下联系电话。
如需发票，请在汇款单附言注明“发票”以及发票抬头，过后将不能补开。
本活动仅限于在杂志社订阅的读者，邮局订阅等其他渠道不参加此活动。
由于本次活动涉及奖品发放，参与活动的读者将不能中途退订。
**邮费：**平寄邮费由杂志社承担，如需挂号，每本另加3元挂号费，汇款时一并汇上，并注明挂号字样。

2013 年第 15 期
8 月(上)

# 电脑编程技巧与维护

总第 285 期　　1994 年 7 月创刊　　(半月刊)


地　　址：北京市海淀区长春桥路 5 号
　　6 号楼 1209 室
投 稿 信 箱：gaojian@comprg.com.cn
　　gaojian@comprg.sina.net
编辑部信箱：gaojian@comprg.com.cn
发行部信箱：zzsfx@vip.sina.com
网　　址：http://www.comprg.com.cn
邮　　编：100089
电　　话：(010) 82561037
传　　真：(010) 82561614
照　　排：《电脑编程技巧与维护》
　　杂志社电脑排版部
印 刷 厂：北京慧美印刷有限公司
订 阅 处：全国各地发行局
国内总发行：北京报刊发行局
邮 发 代 号：82-715
国外发行代号：M6232
刊　　号：ISSN 1006-4052 / CN11-3411/TP
广告订可证：京海工商广字 0151 号
全 年 定 价：264 元
每 期 定 价：11 元

# 电脑编程技巧与维护(半月刊)

每月3日出版


实用第一 智慧密集

# 目次





实用第一 智慧密集

## Linux 基金会发布开源虚拟机管理程序 Xen 4.3

最近，Linux 基金会组织发布了 Xen 虚拟机管理程序的新版本：Xen 版本 4.3，准备好了兑现其承诺。

据该组织声称，Xen 4.3 是历时九个多月开发的工作成果，1362 个变更集 (changeset) 当中包括对 136128 余行代码所作的变更——这归功于来自 27 个不同组织和 25 家独立个体开发商的 90 个人。思杰仍贡献了最大比例的变更集，达到 41%，而其他组织同样挺身而出，比如 Suse Linux 贡献了 23%，英特尔贡献了 6%，美国国家安全局 (NSA) 贡献了另外 5%。

新版本采用了 GPLv2 许可证，包括许多新功能和改进之处，比如以下：

●性能和可扩展性方面的改进：新的非一致内存访问 (NUMA) 调度器将在 NUMA 硬件上提供大幅改进性能的优点。Xen 4.3 还将主机上支持的物理内存数量从 5TB 增加到了 16TB。去除了最多 300 个虚拟处理器的工具堆栈瓶颈限制，现在测试表明支持的虚拟处理器达到 750 多个。由于块协议方面具有可扩展性，用户们会看到读/写性能和吞吐量有所提升，单个主机上可以有 6 个以上的访客。

●经过改进的 QEMU 集成：之前版本的 Xen 一直使用快速仿真器 (QEMU) 硬件仿真器的分支版，但是 Xen 项目已将默认的 QEMU 改为 QEMU-XEN，该仿真器基于上流 QEMU 项目的版本。这样一来，Linux 操作系统的发行者就更容易把 Xen 集成到各自的发行版中了。

●集成软件定义网络 (SDN)：Xen 包括 Open vSwitch 技术的技术预览，该技术由现隶属 VMware 公司的 Nicira 开发而成。Open vSwitch 是一项开源虚拟交换机技术，充当一种桥接机制，以取代起初是 Xen 一部分的较旧的虚拟接口代码。预计这将是 Xen 4.4 版本中一项全面得到支持的功能。

●更高的电源效率：Xen 现在使用所有现代英特尔处理器 (由 Sandy Bridge 开始) 的 MWAIT 扩展，只要处理器支持 MWAIT 扩展。这有望改进 Xen 的电源效率。

Xen 4.3 中最值得关注的新功能也许是“技术预览”(Technolgy Preview)，即为基于 ARM 的处理器虚拟化提供试验性支持。这种支持不仅包括目前的 32 位版本，还包括将随 ARMv8 架构一同推出的即将发布的 64 位版本。

思杰公司的软件工程师 George Dunlap 在 Xen.org 的一篇博文中表明，针对 32 位版本移植的 Xen 目前可在 ARMv7 快速模型 (ARMv7 Fast Models) 上启动，该模型包括 Cortex A15 平台。Dunlap 写道，在该架构上，“Xen 可以启动 dom0，创建其他虚拟机，并且支持虚拟机生命周期的所有基本操作。”尽管目前市面上还没有面向 64 位 ARM 片上系统 (SoC) 的硬件，但 Dunlap 表示，工程师们还设法让 Xen 在 ARM 的 ARMv8 实时系统模型上在 64 位模式下顺畅地运行。

Xen 项目的社区经理 Lars Kurth 补充说：“支持 ARM 服务器对开源社区而言是个令人激动的进展;我们为这给客户们带来的种种机会而感到兴奋。”

## 微软推出 Lab of Things：支持多人云端共用物联网

据报道刚不久，微软研究 (Microsoft Research) 推出了测试版的 Lab of Things，这是一个新的平台，以一种简单的方式支持从物理世界收集的传感器信息，允许多人在不同的地方进行各种实验。

Lab of Things 是一个将物理数据收集和微软的 HomeOS 连结在一起的系统。HomeOS 是微软试图将家庭变成更为自动化、智能化的系统。简单而言，如果你想要利用传感器数据进行一项实验，Lab of Things 将提供一个简单的终端。你甚至可以通过移动设备接入你的实验、通过云端存储和分享数据、使得数据自我修正、还能够参观实验地点。

举例而言，如果你想要做一个实验，这个实验需要收集你所在的城市的无线电塔顶端的气温数据。一旦你在电塔顶端安装了传感器，并将这些数据传入网络，你就可以利用 Lab of Things 作为工具去收集、管理以及分析这些数据。这就像蝙蝠侠的移动犯罪数据库一样，但 Lab of Things 则是真实物理世界的东西。

微软将这个服务成为“几乎是即时”的服务。Lab of Things 的一大优势在于将好想法和实验之间的距离缩短，不再需要科学家，也无须像软件工程师那样自我编程来做测试自己的一些想法——而只需要一些代码就能进行实验。这会使得实验更加的简便快捷，同时也降低了实验的资金依赖——DIY 的黑客们如今可以更快的钻研现实世界的数据。这个服务是大数据、云端、软件服务以及物联网的汇流。

## 谷歌收购语音识别专利对抗 Siri

美国技术咨询公司 (SR Tech Group) 日前宣布，谷歌已经向该公司购买了多项专利，涵盖搜索引擎语音接口和调整语音识别程序的相关技术。

SR Tech Group 与总部位于辛辛那提的 VoiceTechGroup 展开了合作，后者开发了众多语音识别技术。

谷歌曾在今年五月的 I/O 开发者大会上展示了“对话搜索”功能。对话搜索指的是用户可以直接向搜索引擎发问，而不必手动输入搜索条目。要使用这项工具，用户需要借助 Chrome 浏览器和麦克风。

谷歌已经展开了不少投资来抗衡苹果公司的 Siri 语音助理。谷歌专利事务副总法律顾问阿伦·罗 (Allen Lo) 在声明中说：“谷歌赞赏 SR Tech Group 的创新，很高兴收购 SR Tech Group 的专利和专利申请。这些专利组合对我们现有的 5 万多项专利和专利申请形成了补充。”

# 编程岁月

汪永松

**作者简介**

汪永松（1980年10月－），2012年6月毕业于武汉大学测绘工程专业，硕士。具有多年IT行业开发及管理经验，现为北京某IT公司售前顾问。著有《J2ME手机高级编程》、《Android平台开发之旅》等书。目前主要关注Web开发框架、移动客户端、嵌入式数据库等方面的研究及应用。

计算机改变了我的人生轨迹：首先是从财会专业跨入到计算机行业，再是从普通的程序员逐步炼成技术骨干、项目经理。

可能和大多数程序员不一样，我是跨行学计算机，同学之间对编程的交流甚少，加之当时上互联网也不方便，所以很多内容都是通过购书学习和上机摸索积累而来，那些初学时的艰涩、调试bug时的孤寂至今还让笔者无法释怀。

在一头雾水地接触编程的时候，加上“编程就是吃青春饭”的传言纷纷，我也曾纠结过何去何从。幸运的是，还是找到了用计算机去解决问题和去改变的兴趣，兴趣如同扬起帆的船，跨越一个又一个难题，闯入一个又一个世界；其间那些“山穷水尽疑无路，柳暗花明又一村”的欣喜，如一座座闪亮的灯塔，鼓舞前行。

## 1 蓬门今始为君开

最初接触的开发工具是FoxBASE，记得是在DOS界面下，无需编译成可执行文件，通过命令行进行交互执行，可以非常方便地创建数据库表，进行记录的查询、新增、删除和更新操作。从现在的经历来看，FoxBASE算是比较简陋，甚至谈不上是专业的开发环境。但毋庸置疑，它毕竟还是充当了我编程生涯的启蒙老师。对于变量的定义、程序结构、模块化、SQL，这些概念和认识却一直延续到现在，只不过对它们的认识是逐步加深和拓宽。

大三的时候开始学习C语言，采用的教程是清华大学出版社出版的《C程序设计》（谭浩强编著，第1版），上机使用的是Turbo C（又戏称为拖把C）。相比FoxBASE，C语言是真正的编程语言，但笔者认为它不是一个完整的开发环境，而不像Linux平台下的C编程，包含了网络通信、多线程等内容。从专业化的角度而言，C语言不仅强化自己在编程方面的概念，更是夯实了编程的思维模式，无论是从数据定义、表达式运算、指针应用，还是从函数设计、文件操作、宏定义，都有助于举一反三地应用到其他语言（例如后接触的JavaScript和Java）。

对开发者而言，C语言需要关注的细节较多，例如：数据在内存中的存储状态、操作符的优先级、堆内存的分配和回收，这些都是让初学者畏难和晦涩难懂的地方，在开发的初期阶段可能会觉得毫无乐趣可言。尽管如此，建议程序员还是应该把C语言当成一门帮助自己修炼“内功”的编程语言，只有经过了它的严格历练，程序员就会养成了谨小慎微、规范清晰的编程习惯，以至于转入到其他的开发环境，或是其他语言的开发中，都会觉得得心应手，游刃有余。

深入了解了C语言，倦腻于DOS开发环境，开始渴望可视化编程，希望能够编制出如同Windows程序那样具有丰富多彩的界面效果、能够响应鼠标和键盘的执行程序。基于此，如饥似渴地接触C语言的图形库绘制以及高级DOS编程。从求伯君编写的一本有关DOS编程的书籍中，了解了如何使用C语言的DOS库调用系统中断，从而实现例如：磁盘、鼠标、屏幕功能等系统功能。然后一段时间的激动之后发现，虽然可以自行绘制各种形状和颜色的界面，但是这些和Windows程序的效果和操作比起来还是有天壤之别，一下子我开始沮丧起来。

## 2 无心插柳柳成荫

一次偶然的机会，笔者无意之间翻看了同学借的一本有关JavaScript的书，结果发现可以使用JavaScript进行可视化编程。以现在的认识而言，JavaScript是一种宿主脚本语言，其属于Web开发的范围，但当时对我而言，JavaScript可以用来编制网页程序，使用记事本就可以写代码，再使用浏览器加载运行，开发环境简单明了，而且JavaScript的语法也颇类似于C语言，所以学起来还比较得心应手，对可视化编程可谓是踌躇满志。还记得当我用JavaScript在网页上弹出一个对话框时的惊喜，应该算得上是我人生中写出的第一个可视化对话框。特别值得一提的是，JavaScript语言中的对象、属性、方法、

事件等概念，成了笔者面向对象编程的启蒙，帮助我后来平稳地从 C 过渡到 C++。

打从 JavaScript 起，再到 HTML、CSS，我开始领悟了可视化编程的一些核心概念，例如：可视组件（Web 元素）的控制、事件（鼠标、键盘、窗口）的响应、对象创建和引用等。所以至今还感谢那一次与 JavaScript 的偶遇，不仅帮助我开启了可视化编程的大门，更是把笔者带入了面向对象编程的世界。有趣的是，当时我对 Web 后端开发却没有太多感应，而是等若干时间后才再次接触。

## 3 乱花渐欲迷人眼

继 JavaScript 之后，开始关注和寻找 Windows 编程的教程，我使用的第一款 Windows 程序开发环境是 Borland C++ Builder (5.0)，当即被其快速开发的模式给震撼了：基于组件的窗体设计、消息回调机制、数据库编程、网络通信、三层架构、Web 开发以及整套体系的 Win32 API 参考。一时间，我就像一个小孩打开了装满礼物的包裹，看到映入眼睛全是琳琅满目的新鲜玩意，欣喜若狂。为了更好地学习 C++ Builder，我买了一本也迄今为止买的最厚的一本有关 C++ Builder 程序设计的高级教程，有近 750 页，台湾人写的。当时的情形是，一边看书一边上机验证，验证不了的再反过来看书，再次揣摩书中的细节，再次上机验证，最后把上机的总结在书中进行标注。

由衷地说，C++ Builder 是一款优秀的集成开发环境，其几乎包含了可视化程序开发的所有功能范围，正是它把我引入到职业编程的轨道上。C++ Builder 采用的 C++语法，而从 C 到 C++，JavaScript 是功不可没，面向对象和对类的理解，都是由它借鉴而来。随着对 C++ Builder 的理解和使用的不断加深，我开始摸索 Windows 编程的高级技巧，例如：自定义组件、三层架构数据库应用、多线程、并发控制、分装动态链接库等。然而，其中一个不算很起眼开发功能却把笔者引入到另外一个天地，即 Web 开发。

C++ Builder 5 支持的是 CGI 和 ISAPI，Web 后台调用这些模块可以访问数据库并生成网页内容，数据的显示效果只需要与 HTML 融合即可，且客户端无需分发程序，而不像独立程序那样，数据的组织和展示组件都必须编码，而且程序的发布需要复制分发。鉴于 Web 程序如此神奇的开发模式，我开始迷上了 Web 后端开发，而且一发不可收拾。曾在半年的时间里，先后尝试了 PHP、ASP、JSP 等服务页的编写和部署，其中 PHP（3.0）简单易用，就是语法比较怪异（相对 JavaScript 和 C++，其类似于 Perl），同样，笔者对 ASP 也不怎么待见，而倒是钟情于 JSP，觉得其中的 Java 语法和 C++、JavaScript 很接近。尽管如此，我对 JSP 的兴趣没有维持多久，其原因竟然是觉得 Tomcat 不好用，而觉得不好用的原因，经过后来分析，主要原因有两点，一是对 XML 配置不熟悉，再是对类容器的概念尚未形成。这一段插曲算是与 Web 开发的第二次接触，结果是浅尝辄止。其间，我还学习了数据结构和算法，主要是为软考做准备。

## 4 绝知此事要躬行

从 2003 年开始自己的 IT 职业，主要使用 C++ Builder 进行开发，主要开发 MIS 系统。在一年多的项目实践中，先后使用过 Access、SQL Server、Oracle 等数据库，在数据库的设计、数据的层次设计及 SQL 的编写技巧方面积累了不少开发经验。其中参与了一个硬件通信的小项目，却得到了意想不到的收获。该项目是通过工控机的串口获取多个终端传来的信号信息，然后在界面上实时显示信号，此外还包括向各个终端发送控制信号。通过这个项目，掌握了在 I/O 控制中的多线程编程以及其间的并发控制，也为后续研究 HTTP 服务器做了很好的铺垫。与此同时，我开始自学 Java，并尝试写一个 Java HTTP 服务器，这个服务器实际上就是一个类容器，采用多线程的方式，按照 HTTP 标准接收和返回客户端的请求和回复，支持 CGI 进程的调用和类似于 JSP 页面的解析。这个尝试，最终激起了我对 B/S 架构的兴趣，并延续到现在。

不到两年时间，由于工作的原因，我改用微软公司的 Visual C++工具进行开发，主要是开发数据处理系统。一方面，了解 MFC 类库，使用 ADO 访问 SQL Server 数据库等，并开始接触 STL 等标准库或外部库，开始有意识地使用或积累一些功能稳定、简单易用的工具库或功能库。另一方面，依旧保持自学 Java 的态势，开始摸索 JDBC、XML、RMI 等技术的应用，并对开源产生兴趣，并尝试引入到实际项目当中。自此我的 IT 生涯的两条技术主线：C/C++和 Java 基本建立，C/C++的应用主要还是 C/S 框架应用、桌面程序和基础库；而 Java 的视野则更为广泛，除了桌面程序和 B/S 架构应用，还引发到移动应用。

## 5 柳暗花明又一村

自从在诺基亚手机（S40）上用 J2ME 写了第一个程序后，我就一发不可收拾，从界面到无线通信，从多媒体开发到个人信息管理，差不多花了大半年的时间对 J2ME 手机开发进行了全面的尝试，并于 2008 年将所开发的案例编写成《J2ME 手机高级编程》一书，交由机械工业出版社出版。应该是说尝到了手机移动开发的甜头，加之 J2ME 已经有点暮气沉沉，于是在 J2ME 之后我迅速转移到 Android 平台的开发当中。相比 J2ME 平台，Android 的架构更为系统和完整，几乎不受到制约，其信息获取渠道和数据处理能力更为强大，例如：电话系统、数据库、核心系统服务等内容都是 J2ME 平台没法比拟。在对 Android 平台开发了解得差不多时，发现 Android 程序开发的范围可以延伸到本地（Linux），即通过本地开发库（NDK）的方

式，使用 C/C++语言编写本地库（so），然后使用 JNI 的方式在 Android 程序端进行调用。也就是说开发一个 Android 程序可以将 Java 和 C++两种语言进行结合应用。后来经过摸索发现，很多多媒体应用的程序都是采用如此模式：后端使用 C++库对媒体数据进行编码/解码，而在前端使用摄像头或渲染容器捕获或展示媒体数据。

Android 平台让我第一次觉得有点天地合一的感觉，不仅可以整合使用 C++和 Java 开发语言，而且还可以集成 B/S 应用（基于 Wi-Fi 或 3G 网络）和 C/S 应用（WLAN 通信）。

## 6 似曾相识燕归来

再次由于工作的原因，真正转入到 Web 开发的，主要是面向基于 J2EE 平台的商业智能（BI）应用。从应用的广度而言，前期使用 C++主要关注的是后台功能，例如：数据抽取、解析、转换和存储，而现在面对的 Web 应用不仅关注功能而且还关注数据的集成和展现，从数据存储到组织、传输、解析，再到集成展示，几乎涵盖了我从事 IT 行业以来的所有业务环节。从应用深度而言，Web 应用更突出其服务特性，数据的处理都围绕服务的提供而展开。

这一次，我不仅重新拾起了曾经浅尝辄止的 JavaScript、HTML、CSS、JSP、Tomcat，而且还学会了吸收开源项目或外部库，学会使用或者搭建开发框架。而这些行为模式的转变，估计还是与 Web 开发的特殊性有关。当笔者对 Web 开发了解到一定程度时，Web 开发给笔者的印象是“三多”：标准多、框架多、资源多。所以，在学习和摸索的过程中，需要注重总结和积累，注意引进和复用，把更多的精力转移到业务处理。

作为新接触的环节，在数据集成和展示方面，我也逐步找到适合的资源和模式。在数据集成方面，曾尝试在 Web 后端和前端之间添加一个数据服务层，数据服务层与后端和前端页面均采用 JSON 格式进行交互；当前端页面通过 Ajax 的方式进行异步数据请求，数据服务层将 Web 后端获取的数据以 JSON 的格式反馈给请求页面，从而实现异构数据的集成。在数据展示方面，可以使用 Flex 等富客户端框架绘制专业性较高图表，也可以使用基于 JavaScript 的展示组件库来快速实现常规界面（例如：表格、树状组件等）。

幸运地说，重新对 Web 开发的接触和实践，使得我对数据处理业务的理解更加全面和系统化，而且从开发模式上，实现两个转变：从 C++时期的主要“依靠自己开发”到 Web 时期主要的“整合现有的来开发”的转变；实现了从使用固定的开发环境，到自己构建开发框架的转变。

至此，从技术体系稍作归纳，目的是为了让读者清晰地了解我应用的技术范围，主要包括 4 个方面：C（C/C++）、J（Java）、E（嵌入式）和 W（Web），如图 1 所示。其中 C/C++主要是桌面开发（Windows、Linux）；Java 主要包括桌面开发和开源技术；嵌入式主要包括 J2ME 和 Android；Web 开发主要是基于 J2EE 平台的开发。

对于开发语言或环境而言，我感觉受到 C++ Builder 的影响，以至于刻意规避选择其他的开发环境和语言。然而，作为软件开发者，如果需要保持自己技术的先进性，就必须正视陈出不穷的技术和产品，保持学习和摸索的心态；还必须善于总结和积累当前的技术和产品，持续改进并实现模式化。一个职业的开发者如果能够做到这两点，这不会混乱于当前，不畏惧于今后。这也意味着，开发者必须花费更多的时间来关注行业动态，注入更多的心血来提炼自己的工作，确定编程是自己兴趣所致，则无怨无悔。

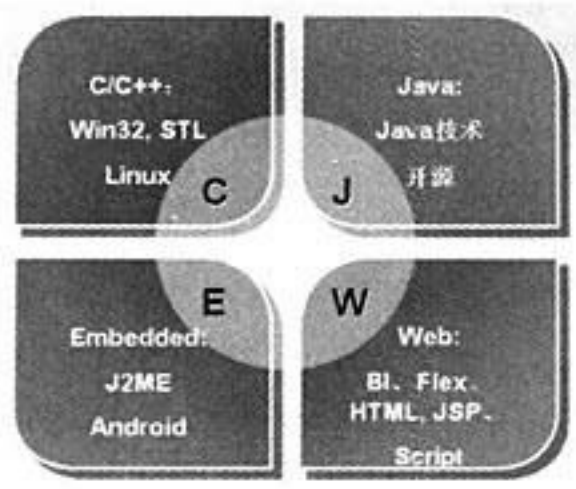



图 1 技术体系示意图

人生如戏，岁月如歌。程序员使用 IT 技术造就了世界的丰富多彩，而自己的人生也被充满酸甜苦辣。回首自己的编程岁月，如同品茶一般，恐怕除了自己，没有谁能够真正体会其中的味道。

### 2013 IBM 软件技术峰会召开

2013 IBM 软件技术峰会于 7 月 11 日-12 日在京召开，《大数据时代》作者维克托·迈尔·舍恩伯格将进行主题演讲。作为行业引领者，IBM 一直走在新趋势的前沿，对于大数据更是全力融合创新技术与行业经验助力客户梳理大数据概念，选择战略方向，制定方案策略，实现行业落地。

IBM 全球副总裁兼大中华区软件集团总经理胡世忠在 IBM2013 技术峰会（IBM Tech Summit 2013）上发表演讲，他表示：“最新的 IBM 全球 CEO 调研显示全球企业领导者将技术列为重要的外部力量，而‘科技史第一生产力’一直被广大中国的企业家和技术人们奉为至理名言。”IBM 认为，在由新一代技术组成的智慧运算时代，中国的企业家们需要更为战略地思考信息科技的定位，将其运用到自身的变革转型中。

IBM 正在通过业务前线以客户为中心的业务转型，通过企业内部全面整合大幅度优化流程和运营，充分发挥最新的科技力量，携手合作伙伴帮助中国企业、组织和政府共同打造‘再现代化’发展的全新方式。

# 利用语义网技术实现铁路交通的地理语义查询（三）

## ——集成本体推理结果并在地理空间中可视化

董 志

摘 要：介绍了在 C# 中集成本体推理功能并解析推理结果，得到火车站换乘查询的语义查询结果，将查询结果进行地理空间中的可视化，以点、线、面的形式在 Web 电子地图中显示。

关键词：语义网；Web3.0 标准；语义查询；功能集成；C# 控件；Web 电子地图

## 1 使用百度地图 JavaScript API 构建应用电子地图

Baidu 地图 JS API 也是一套由 JavaScript 语言编写的应用程序接口（注册后使用），提供了构建地图基本功能的各种接口，提供了诸如本地搜索、路线规划等数据服务，能够在 HTML 中方便地构建电子地图应用，显示效果如图 1 所示。

图 1 Baidu 地图应用效果

本功能程序由 JS 编写，C# 的 WebBrowser 调用，桌面程序与 JS 程序进行交互，显示地图及添加覆盖物。函数 loadTrainLine 装载了影像电子地图，同时根据桌面传递的本体推理结果创建铁路站点的点、线、面等 Overlay 对象，点表示站点，前后站点连接成线，然后将站点所在城市的行政区域面添加到电子地图上去，具体代码如下：

```
<! DOCTYPE html>
<html>
<head>
<meta http –equiv = "Content –Type" content = "text/html;
charset=utf–8" />
<style type="text/css">
body, html,#allmap {width: 100% ;height: 100% ;overflow:
hidden;margin:0;}
#l –map {height:100% ;width:78% ;float:left;border –right:2px
solid #bcbcbc;}
#r–result{height:100%;width:20%;float:left;}
</style>
<script type = "text/javascript" src = "http://api.map.baidu.com/
api?v=1.4"></script>
<title></title>
</head>
<body>
<div id="allmap"></div>
</body>
</html>
<script type="text/javascript">
var map;
window.onload = function () { loadTrainLine("","); };
functionloadTrainLine(pointsStr, lineInfo) {
if (map == null) {
        map = new BMap.Map ("allmap", { mapType:
BMAP_HYBRID_MAP});          // 创建 Map 实例
map.addControl(new BMap.NavigationControl());
map.addControl(new BMap.ScaleControl());
map.addControl(new BMap.OverviewMapControl());
map.addControl(new BMap.MapTypeControl());
    }
map.clearOverlays();
map.centerAndZoom("北京", 12);    // 初始化地图,设置中心点
//坐标和地图级别。
map.enableScrollWheelZoom();    //启用滚轮放大缩小
varbdary = new BMap.Boundary();
varpointsStrArr;
    pointsStrArr = pointsStr.split(",");
if (pointsStrArr.length> 1) {
var points = [];
varlineInfoArr = lineInfo.split(",");
for (var i = 0; i <pointsStrArr.length; ) {
var point = new BMap.Point (parseFloat (pointsStrArr [i+2]),
parseFloat(pointsStrArr[i+3]));
if (i > 1) {
varmidPoint = new BMap.Point((parseFloat(pointsStrArr[i+2])
+ parseFloat(pointsStrArr[i–2])) / 2, (parseFloat(pointsStrArr[i+
```

FOLLOW MASTER PROGRAM

```
3]) + parseFloat(pointsStrArr[i-1])) / 2);
            varmyLabel = new BMap.Label(lineInfoArr[i/4-1],
//为 lable 填写内容
                                        {position: midPoint
                    });          //label 的位置
myLabel.setTitle(lineInfoArr[i/4-1]);     //为 label 添加鼠标提示
map.addOverlay(myLabel);
                }
points[i / 4] = point;
var marker = new BMap.Marker(point); // 创建标注

map.addOverlay(marker);              // 将标注添加到地图中
marker.setAnimation(BMAP_ANIMATION_BOUNCE); //跳动
//的动画
varmyLabel = new BMap.Label(pointsStrArr[i],     //为 lable 填
//写内容
                                        {position: point
                                    });
myLabel.setTitle(pointsStrArr[i]);     //为 label 添加鼠标提示
map.addOverlay(myLabel);
bdary.get(pointsStrArr[i+1], function (rs) {       //获取行政区域
var count = rs.boundaries.length; //行政区域的点有多少个
for (var i = 0; i < count; i++) {
            var ply = new BMap.Polygon (rs.boundaries [i], {
strokeWeight: 2, strokeColor: "#ff0000" }); //建立多边形覆
//盖物
ply.setFillOpacity(0);
map.addOverlay(ply); //添加覆盖物
map.setZoom(6);
                }
            });
            i = i + 4;
        }
var polyline = new BMap.Polyline (points, { strokeColor: "
blue", strokeWeight: 6, strokeOpacity: 0.5 });
map.addOverlay(polyline);
      map.centerAndZoom (points [parseInt ((points.length -1)/
2)], 6);
      }
else {

varmyGeo = new BMap.Geocoder();
        // 将地址解析结果显示在地图上,并调整地图视野
myGeo.getPoint("武汉大学", function (point) {
if (point) {
map.centerAndZoom(point, 16);
map.addOverlay(new BMap.Marker(point));
    }
        }, "武汉市");
      }
  }
</script>
```

## 2 C# 主界面中的下拉提示列表

C# 程序中类 StopNameList 用于输入城市名称时进行提示，当用户输入城市名称时可以根据输入字符顺序排列城市的名称提示，从起始字符开始匹配的名字排在前面，其他匹配情况的名字排在后面，如输入“南”，则“南京、南宁、南昌”排在前，“济南”排在后。如果提示程序检测不是城市检索，则自动检索火车站点，代码具体实现如下：

```
...
usingSystem.Data;
usingSystem.Xml;
usingSystem.Collections;
usingSystem.Text.RegularExpressions;
...
classStopNameList
  {
privateToolStripTextBox _attachTB = null;
privateListBox _lb = null;
publicToolStripTextBoxattachTB
      {
get { return _attachTB; }
set { _attachTB = value; }
      }
publicListBoxlb
      {
      get { return _lb; }
      set { _lb = value; _lb.Click += new System.EventHandler
(this.ListBox_Click); }
      }
      private void ListBox_Click(object sender, EventArgs e)
      {
          _attachTB.TextBox.Text = _lb.Text;
          _lb.Hide();
      }
      public void ReAdjustPostion()
      {
      if (_lb!  =null)
          {
              _lb.Left = _attachTB.TextBox.Left;
                  _lb.Top = _attachTB.TextBox.Top +
_attachTB.TextBox.Height + 20;
                  _lb.Width = _attachTB.Width;
          }
      }
      public void showContent()
      {
      if (_lb ! = null && _attachTB ! = null)
          {
      ReAdjustPostion();
              _lb.Items.Clear();
```

```
SQLiteDBsqliteDbObj = new SQLiteDB();
sqliteDbObj.openConnection("");
                DataSetdb = sqliteDbObj.getRSFromDb
("select cityName from capitalOfprovince where cityName
like '" + _attachTB.Text.Trim() + "'||'%'");
db.Merge (sqliteDbObj.getRSFromDb ("select cityName
from capitalOfprovince where cityName like '% ' ||'" +
_attachTB.Text.Trim() + "' and cityName not like '" + _attach
TB.Text.Trim() + "'||'%'"));
//如果不是城市检索,则检索站点
if (db.Tables[0].Rows.Count<= 0)
                {
 db.Merge (sqliteDbObj.getRSFromDb ("select qid as
cityName from trainStop where qid like '" + _attachTB.Text.
Trim() + "'||'%' group by qid"));
db.Merge (sqliteDbObj.getRSFromDb ("select qid as
cityName from trainStop where qid like '%'||'" + _attachTB.
Text.Trim() + "' and qid not like '" + _attachTB.Text.Trim() + "
'||'%' group by qid"));
                }
if (db.Tables[0].Rows.Count> 0)
                {
for (int i = 0; i <db.Tables[0].Rows.Count; i++)
_lb.Items.Add(db.Tables[0].Rows[i][0].ToString());
                _lb.Visible = true;
                }
else
                {
                _lb.Visible = false;
                }
            }
        }
    }
```

## 3 解析语义查询结果并集成 DeskTop 和 Web 程序

C# 程序中还需要一个本体查询结果处理类，该类解析 RDF 结果，生成字符串传到 HTML 页中的函数中去。则设计类 ParseChangeStopFromXml 用于解析推理结果，写入主界面中的 DataGridView 控件，并根据用户选择的查询结果生成参数调用 HTML 中的 loadTrainLine 函数，交互 Baidu 地图的显示。这个类同时也集成了程序中各函数功能，并且能够调用 JS 程序：

```
classParseChangeStopFromXml
    {
string _reasonReasonreSultStr = "";
privateDataGridView _busStopDG = null;
privateDataGridView _busLineDG = null;
privateWebBrowser _webBrowserMap = null;
privateArrayList _stopLocationInfoArr = new ArrayList();
publicParseChangeStopFromXml()
        {
        }
public void iniDGV(DataGridViewdg,stringcolNameArr)
        {
string [] tempColNameArrTemp = colNameArr.Split(',');
dg.ColumnCount = tempColNameArrTemp.Length;
 dg.ColumnHeadersDefaultCellStyle.BackColor = Color.
DarkSlateBlue;
 dg.ColumnHeadersDefaultCellStyle.ForeColor = Color.
White;
dg.ColumnHeadersDefaultCellStyle.Font =
new Font(_busStopDG.Font, FontStyle.Bold);
dg.AutoSizeRowsMode =
                DataGridViewAutoSizeRowsMode.
DisplayedCellsExceptHeaders;
dg.ColumnHeadersBorderStyle =
DataGridViewHeaderBorderStyle.Single;
dg.CellBorderStyle = DataGridViewCellBorderStyle.Single;
dg.GridColor = Color.Black;
dg.RowHeadersVisible = true;
for (int i = 0; i <tempColNameArrTemp.Length; i++)
        {
dg.Columns[i].Name = tempColNameArrTemp[i];
                dg.Columns[i].AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill;
        }
dg.SelectionMode = DataGridViewSelectionMode.
FullRowSelect;
dg.MultiSelect = false;
dg.ReadOnly = true;
dg.AllowUserToAddRows = false;
                dg.RowsDefaultCellStyle.WrapMode
= DataGridViewTriState.True;
        }
public void iniBusLineDG()
        {
iniDGV(_busStopDG, "城市火车联通");
iniDGV(_busLineDG, "火车线路换乘");
        }
public string reasonReasonreSultStr
        {
get { return _reasonReasonreSultStr; }
set { _reasonReasonreSultStr = value; }
        }
publicDataGridViewbusStopDG
        {
get { return _busStopDG; }
 set  { _busStopDG = value; _busStopDG.Click += new
System.EventHandler(this.showContentFromBusStopLB); }
        }
publicDataGridViewbusLineDG
        {
```

FOLLOW MASTER PROGRAM

```
    get { return _busLineDG; }
    set  { _busLineDG = value; _busLineDG.Click += new
System.EventHandler(this.showLineTomap); }
        }
    publicArrayListstopLocationInfoArr
        {
    get { return _stopLocationInfoArr; }
    set { _stopLocationInfoArr = value; }
        }
    publicWebBrowserwebBrowserMap
        {
    get { return _webBrowserMap; }
    set { _webBrowserMap = value; }
        }
    public string GetCityNmaeFromXml(string parseXml)
        {
      _busStopDG.Parent.Cursor = Cursors.WaitCursor;
            _reasonReasonreSultStr = "";
            _busStopDG.Rows.Clear();
            _busLineDG.Rows.Clear();
    XmlDocumentxmlDoc = new XmlDocument();
    xmlDoc.LoadXml(parseXml);
  XmlNodeListNodeList = xmlDoc.GetElementsByTagName
("result");
    if (NodeList ! = null)
          {
    try
              {
    foreach (XmlNodestNode in NodeList)
                  {
    XmlDocumentxmlDocTemp = new XmlDocument();
    xmlDocTemp.LoadXml(stNode.OuterXml);
                XmlNodeListNodeListTemp = xmlDocTemp.
GetElementsByTagName("literal");
          //通过 Attributes 获得属性名字为 name 的属性
                    string reachStr = " 换乘次数:" +
(NodeListTemp[0]).InnerText;
    if ((NodeListTemp[0]).InnerText == "0")
    reachStr += ";直达线路:";
    if ((NodeListTemp[0]).InnerText == "1")
    reachStr += ";1 次换乘线路:";
     reachStr += "" +  (NodeListTemp [1]).InnerText + "," +
(NodeListTemp  [2]).InnerText + "," +  (NodeListTemp [3]).
InnerText + "," + (NodeListTemp[4]).InnerText;
    reachStr = reachStr.Replace(",,", ",").Replace(",,", ",");
    string[] row = {reachStr};
                    _busStopDG.Rows.Add(row);
                    _reasonReasonreSultStr += reachStr;
                }
            }
    catch (Exception e)
            {
Console.Write(e.Message);
            }
showContentFromBusStopLB(this, new EventArgs());
}
        _busStopDG.Parent.Cursor = Cursors.Default;
return _reasonReasonreSultStr;
    }
private void addBusLineToDG(DataRowdr, int type)
    {
            string listItemStr = " 直达线路:1. " + dr["qid"].
ToString() + "->" + dr["zid"].ToString();
    listItemStr += " 车次:" +dr["cid"].ToString()+" ";
    if (dr["yz"].ToString().Length > 0)
    listItemStr += " 硬座价格:" +dr["yz"].ToString();
    if (dr["rz"].ToString().Length > 0)
    listItemStr += " 软座价格:" +dr["rz"].ToString();
    if (dr["yw"].ToString().Length > 0)
    listItemStr += " 硬卧价格:" +dr["yw"].ToString();
    if (dr["rw"].ToString().Length > 0)
    listItemStr += " 软卧价格:" +dr["rw"].ToString();
    if (dr["yd"].ToString().Length > 0)
    listItemStr += " 动车一等坐:" +dr["yd"].ToString();
    if (dr["ed"].ToString().Length > 0)
    listItemStr += " 动车二等坐:" +dr["ed"].ToString();
    if (type > 0)
        {
                string[] row = { _busLineDG.Rows
[_busLineDG.Rows.Count  -  1].Cells  [0].Value.ToString ().
Replace("直达线路:", "换乘线路:") + listItemStr.Replace("直达
线路:1.", "2.") };
            _busLineDG.Rows [_busLineDG.Rows.Count
- 1].SetValues(row);
        }
    else
        {
    string[] row = { listItemStr + ";" };
            _busLineDG.Rows.Add(row);
        }
    }
    private void getBusStopLoction (string stopName, string
cityName, intstartIndex)
        {
    DataSetdbLocation = null;
    SQLiteDBsqliteDbObj = new SQLiteDB();
    sqliteDbObj.openConnection("");
            dbLocation = sqliteDbObj.getRSFromDb("select
trainStopName,lat,lng  from  trainStopLocationCity  where
trainStopName  =  '" + stopName + "'  and cityName ='" +
cityName + "' ");
            _stopLocationInfoArr.Add(stopName);
            _stopLocationInfoArr.Add(cityName);
    if (dbLocation.Tables[0].Rows.Count> 0)
```

```
            {
    if (startIndex> 0)
                {
    startIndex++;
                    _stopLocationInfoArr.Insert(startIndex*4+
1, dbLocation.Tables[0].Rows[0]["lng"].ToString());
                    _stopLocationInfoArr.Insert(startIndex * 4,
dbLocation.Tables[0].Rows[0]["lat"].ToString());
                }
    else
                {
                    _stopLocationInfoArr.Add (dbLocation.
Tables[0].Rows[0]["lng"].ToString());
                    _stopLocationInfoArr.Add (dbLocation.
Tables[0].Rows[0]["lat"].ToString());
                }
            }
    else
            {
    dbLocation.Clear();
            dbLocation = sqliteDbObj.getRSFromDb("select
cityName,lat,lng from CapitalOfprovince where  cityName='"
+ cityName + "' ");
                _stopLocationInfoArr.Add (dbLocation.Tables
[0].Rows[0]["lng"].ToString());
                _stopLocationInfoArr.Add (dbLocation.Tables
[0].Rows[0]["lat"].ToString());
            }
        }
    public void showContentFromBusStopLB(object sender,
EventArgs e)
        {
    _busStopDG.Parent.Cursor = Cursors.WaitCursor;
    if (_busStopDG ! = null)
            {
        stringshowStopInfo = _busStopDG.SelectedRows[0].
Cells[0].Value.ToString();
    string[] showStopInfoArr = showStopInfo.Split(';');
                _busLineDG.Rows.Clear();
    SQLiteDBsqliteDbObj = new SQLiteDB();
    sqliteDbObj.openConnection("");
    DataSetdbLine = null;
                _stopLocationInfoArr.Clear();
    if (showStopInfoArr[0].Split(':')[1] == "0")
                {
    string [] cityNameArr = showStopInfoArr [1].Split (':')[1].
Split(',');
     dbLine = sqliteDbObj.getRSFromDb ("select cid,qid,zid,
yz,rz,yw,rw,yd,ed,qLocation,zLocation from trainStop  where
qid = '" + cityNameArr[0] + "'  and zid='" + cityNameArr[1] +
"' order by  abs(coalesce(yz,yd))");
    if (dbLine.Tables[0].Rows.Count> 0)
                {
    for (int i = 0; i <dbLine.Tables[0].Rows.Count; i++)
                {
    addBusLineToDG(dbLine.Tables[0].Rows[i], -1);
     getBusStopLoction (dbLine.Tables [0].Rows [i] ["qid"].
ToString(), dbLine.Tables[0].Rows[i]["qLocation"].ToString(), -
1);
     getBusStopLoction (dbLine.Tables [0].Rows [i] ["zid"].
ToString (), dbLine.Tables [0].Rows [i] ["zLocation"].ToString (), -
1);
                }
            }
        }
    if (showStopInfoArr[0].Split(':')[1] == "1")
                {
    string [] cityNameArr = showStopInfoArr [1].Split (':')[1].
Split(',');
     dbLine = sqliteDbObj.getRSFromDb ("select cid,qid,zid,
yz,rz,yw,rw,yd,ed,qLocation,zLocation from trainStop where
qid = '" + cityNameArr[0] + "'  and zid='" + cityNameArr[1] +
"' order by  abs(coalesce(yz,yd))");
    if (dbLine.Tables[0].Rows.Count> 0)
                {
    for (int i = 0; i <dbLine.Tables[0].Rows.Count; i++)
                {
    if (dbLine.Tables.Count> 1)
    dbLine.Tables.RemoveAt(1);
     DataTabledt = sqliteDbObj.getRSFromDb("select a.cid,a.
qid,a.zid,a.yz,a.rz,a.yw,a.rw,a.yd,a.ed,b.lat,b.lng,qLocation,
zLocation from trainStop a, trainStopLocationCity b where
qid = '" + cityNameArr[1] + "'  and zid='" + cityNameArr[2] +
"' order by  abs(coalesce(yz,yd))").Tables[0].Copy();
    dt.TableName = "secondLine";
    dbLine.Tables.Add(dt);
    if (dbLine.Tables[1].Rows.Count> 0)
                {
    intmaxCount = 10;
    if (maxCount>dbLine.Tables[1].Rows.Count)
    maxCount = dbLine.Tables[1].Rows.Count;
    for (int j = 0; j <maxCount; j++)
                {
    addBusLineToDG(dbLine.Tables[0].Rows[i], -1);
    addBusLineToDG(dbLine.Tables[1].Rows[j], 1);
            getBusStopLoction (dbLine.Tables [0].Rows [i]["
qid"].ToString(), dbLine.Tables[0].Rows[i]["qLocation"].ToString
(), -1);
  getBusStopLoction(dbLine.Tables[0].Rows[i]["zid"].ToString
(), dbLine.Tables[0].Rows[i]["zLocation"].ToString(), -1);
            getBusStopLoction (dbLine.Tables [1].Rows [j] ["
zid"].ToString(), dbLine.Tables[1].Rows[j]["zLocation"].ToString
(), -1);
                }
```

```
                    }
                }
            }
        }
showLineTomap(this, new EventArgs());
        }
        _busStopDG.Parent.Cursor = Cursors.Default;
    }
public void showLineTomap(object sender, EventArgs e)
    {
    intshowIndex = _busLineDG.CurrentRow.Index;
    if (showIndex< _busLineDG.Rows.Count)
            {
    stringbusStopLocation = "";
    intendIndexCount = 0;
                if  (_busLineDG.CurrentCell.Value.ToString().
IndexOf("直达线路") >= 0)
    endIndexCount = 2;
                if  (_busLineDG.CurrentCell.Value.ToString().
IndexOf("换乘线路") >= 0)
    endIndexCount = 3;
    for (int i = 0; i <endIndexCount; i++)
                {
    for (int j = 0; j < 4; j++)
     busStopLocation  += _stopLocationInfoArr [showIndex*
4*endIndexCount+(i*4) + j] + ",";
                }
            busStopLocation = busStopLocation.Substring(0,
busStopLocation.Length – 1);
    stringlineInfo = "";
                string regexStr = "车次:(.+?) ";
    string replacement = "$1";
                 Regex r = new Regex(regexStr); //定义一个
//Regex 对象实例
                string input = _busLineDG.CurrentCell.Value.
ToString(); // 在字符串中匹配
     foreach  (Match match in Regex.Matches (input,
regexStr))
                {
    string result = match.Result(replacement);
    if (lineInfo.Length> 0)
    lineInfo += "," + result;
    else
    lineInfo = result;
                }
    dealTainChangeDataToMap(busStopLocation, lineInfo);
            }
        }
      public  void  dealTainChangeDataToMap  (string
busStopLocationInfo,stringlineInfo)
        {
            _webBrowserMap.Document.InvokeScript  ("
loadTrainLine", new String[] { busStopLocationInfo,lineInfo });
//调用 JS 程序
        }
    }
```

## 4　程序运行结果测试

“北京”（模糊）至“武汉”（模糊）直达火车车次模糊查询如图 2 所示。

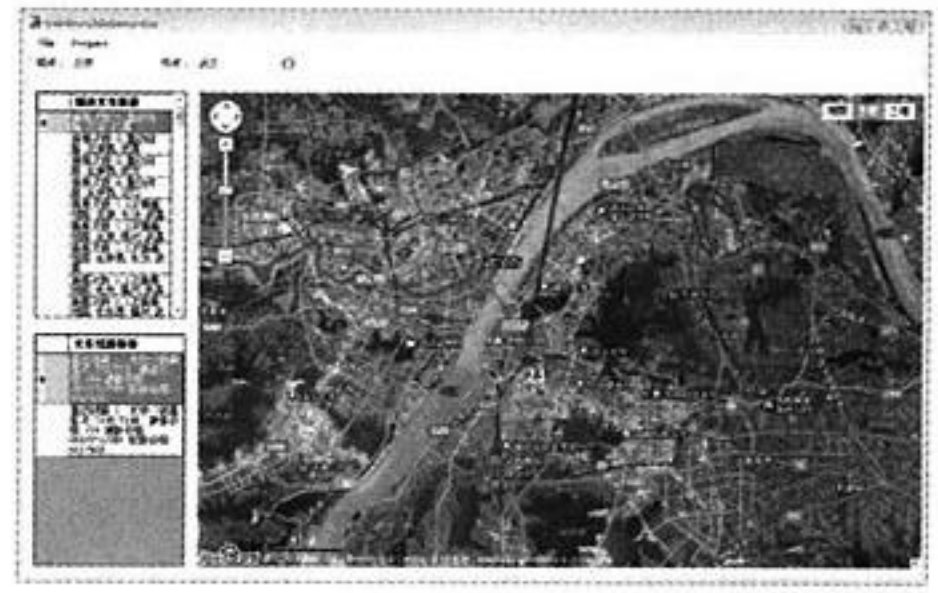


图 2　直达模糊查询结果

“北京”（模糊）至“杭州”（模糊）一次换乘火车车次模糊查询如图 3 所示。

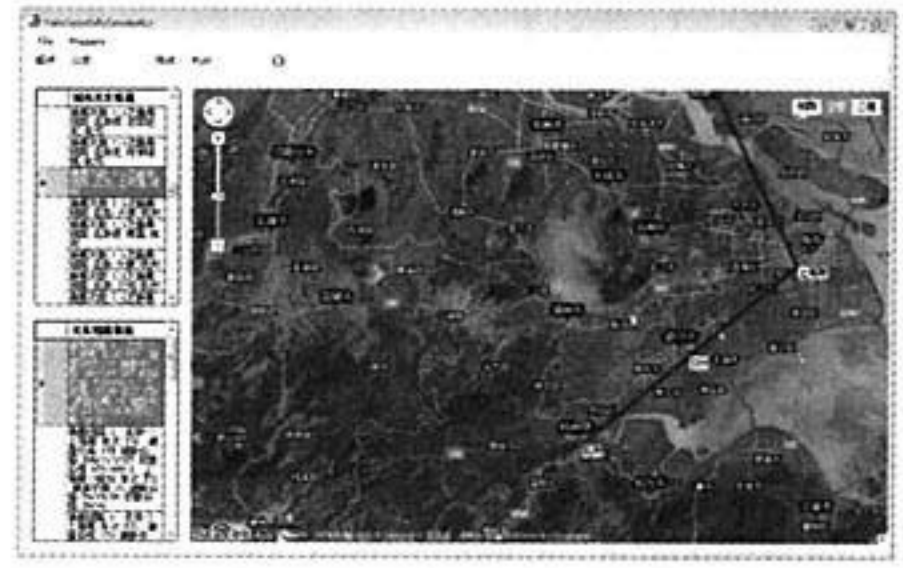


图 3　一次换乘模糊查询结果

“北京”（模糊）至“汉口”（精确）的模糊精确混合查询如图 4 所示。

图 4　模糊精确查询结果

从以上测试结果可知，程序进行语义推理后，可以识别各

（下转第 23 页）

# 自己动手开发开课系统

张贻忠　洪成波

**摘　要**：对一个学生人数较多的学校，排课表是件大事，尤其是象计算机、音乐、舞蹈、体育、美术类课，需要在公用上课地点进行，避免冲突、合理利用资源，这给排课增加了很大难度。用Excel VBA开发一款排课系统，能很好地解决这个难题，让工作简单化。

**关键词**：Excel VBA编程；排课

## 1　设计思路

### 1.1　系统构成

使用6张工作表构成排课系统工作簿，分别为：“系统界面”、“基础设置”、“教师任课”、“各班课表”、“排课模板”、“课表模板”。其中：

(1) “基础设置”工作表，用于设置基础数据，包括：

1）使用单位名称；2）班级编号、班级名称；3）每天节数、每周天数；4）学科编号、学科名称；5）上课地点编号、名称。

(2) “教师任课”工作表，存放教师任课信息和教师个人课表数据，包括：

1）教师编号、教师姓名；2）任课信息（任教班+任教学科+任教节数+上课地点）；3）个人课表数据（班级编号+学科编号+上课地点编号）。

(3) “各班课表”工作表，存放各班课表数据，包括：

1）班级编号；2）班级课表数据（教师编号+学科编号+上课地点编号）。

### 1.2　排课原理

(1) 选择教师；(2) 提取选定教师任课信息（任教班+任教学科+任教节数+上课地点）；(3) 生成该教师任教的所有班课表；(4) 提取这些班已经排入的课程信息；(5) 标记公共上课地点已经被排课的课时；(6) 添加任课代码；(7) 保存排课时，检查各班课时数是否正确，排课是否出现冲突；(8) 将教师任课数据以编号的形式分别存放到“教师任课”和“各班课表”中。

### 1.3　查询和打印课表

选择班级或教师，提取班级课表数据，填入课表模板，并将编号形式转换成名称格式，以便查询打印各班课表或教师个人课表。

### 1.4　系统操作

整个程序使用下拉菜单配合命令按钮，利用用户窗体设置参数，完成操作。

## 2　制作方法

### 2.1　工作表制作

(1) “系统界面”工作表制作。

如图1所示。


图1　系统界面

1）合并及居中D4：J7单元格，黑体、蓝色、48磅，用于显示单位名称，对齐方式为缩小字体填充。2）合并及居中C4:K14单元格，琥珀体、红色、72磅，用于显示系统名称。3）整个工作表填充淡蓝色底色。

(2) “基础设置”工作表制作。

如图2所示。


图2　基础设置

1）把A2:B100、H2:I100、K2:L100设置为允许用户编辑区域，对齐方式为缩小字体填充；2）A列为“班级编号”（第一位：年级编号、二三位：班级序号），B列为“班级名称”；3）D2为“每天节数”；4）E2为每周天数；5）F2为使用单位名称；6）H列为“学科编号”，用01~99，I列为“学科名称”，7）K列为“上课地点编号”，用01~99，L列为“上课地点名称”；8）隐藏C、G、J列；9）在D4:F15区域内输入必要的说明（如图3所示）。

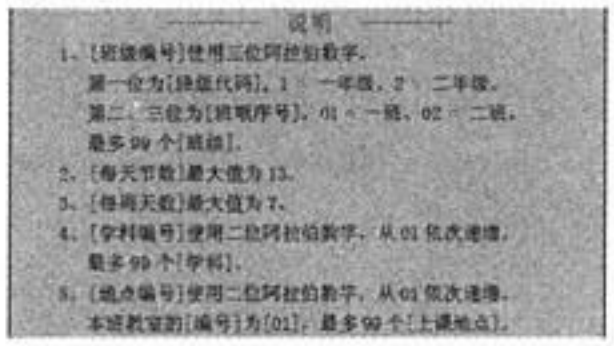



**图 3　工作表制作说明**

(3)　“教师任课”工作表制作。

如图 4 所示。

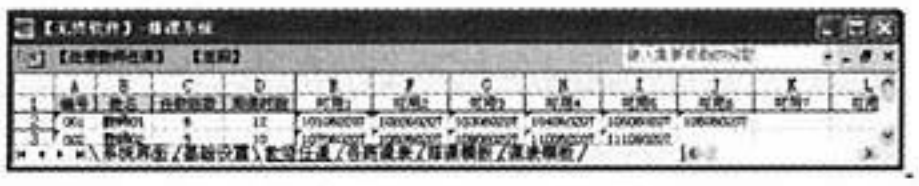


**图 4　教师任课表制作**

1）设置整个工作表为文本型数据，列宽为 9，对齐方式为缩小字体填充；2）A 列为编号（3 位阿拉伯数字），B 列为姓名，C 列为任教班数，D 列为周课时数；3）通过【处理教师任课】下拉菜单，完成“录入任课信息”、“修改任课信息”、“删除任课信息”等操作。

(4)　“各班课表”工作表制作。

如图 5 所示。

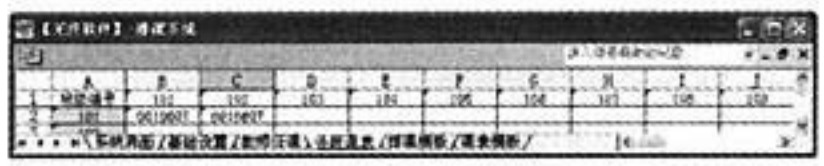


**图 5　各班课表制作**

设置整个工作表为文本型数据，列宽为 9，对齐方式为缩小字体填充，A 列为班级编号。注意：此表中的数据无需输入，都是程序自动添加的。

(5)　“排课模板”工作表制作。

如图 6 所示。

**图 6　排课模板制作**

1）设置整个工作表为文本型数据，列宽为 9，对齐方式为缩小字体填充，淡蓝色底色；2）第 2 行行高为 34；3）合并 K2:O2，黑体、红字、14 磅，加粗框线，填充黄色底，输入文字“提示：用字母 A 代替任课”；4）如图 6 插入“上一人”、“下一人”、“保存数据”3 个按钮。

(6)　“课表模板”工作表制作。

如图 7 所示。

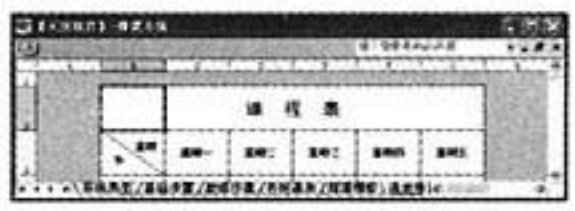


**图 7　课表模板制作**

1）设置整个工作表为文本型数据，列宽为 9，对齐方式为缩小字体填充，淡蓝色底色；（2）第 1 行行高为 20。注意：此表中的数据无需输入，都是程序自动添加的。

## 2.2　用户窗体设计

按图 8 样式设计用户窗体 1，含 9 个 Label、2 个 ComboBox、3 个 TextBox、3 个 CommandButton，各控件属性可使用默认。

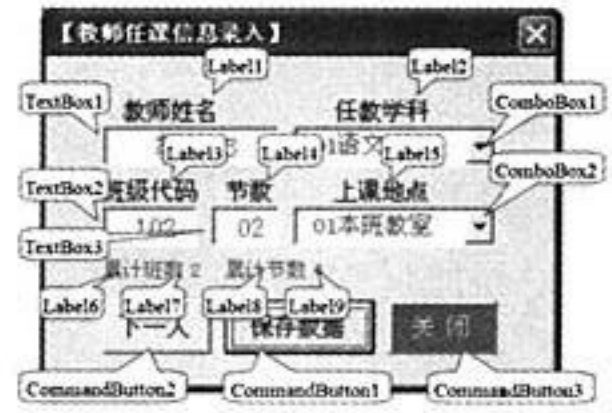



**图 8　教师任课信息录入设计**

按图 9 样式设计用户窗体 2，含 1 个 Frame、2 个 OptionButton、1 个 ComboBox、2 个 CommandButton，各控件属性可使用默认。


**图 9　选择教师窗口设计**

按图 10 样式设计用户窗体 3，共 1 个 ComboBox、2 个 CommandButton，各控件属性可使用默认。


**图 10　提取数据按钮设计**

按图 11 样式设计用户窗体 4，含 1 个 ListBox，控件属性可使用默认。

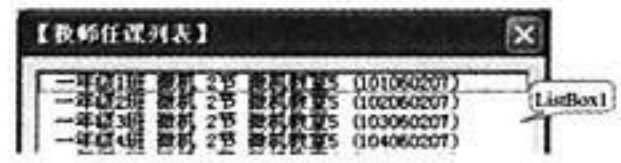



**图 11　教师任课列表**

按图 12 样式设计用户窗体 5，2 个 OptionButton、2 个 ComboBox、3 个 CommandButton，各控件属性可使用默认。


**图 12　课表查询设计**

## 3 程序结构

程序结构如图 13 所示。


图 13 系统结构

## 4 模块代码

【模块 0—启动系统】中的代码：

```
Sub auto_open()
  Application.ScreenUpdating = False ' 关闭屏显
  Application.WindowState = xlMaximized ' 应用程序窗口处
'于"最大化"状态。
  ActiveWindow.WindowState = xlMaximized ' 文件窗口处于
'"最大化"状态。
  Application.CommandBars("Standard").Visible = False  ' 隐
'藏"工具栏"
  Application.CommandBars("Formatting").Visible = False ' 隐
'藏"格式栏"
  Application.DisplayFormulaBar = False    ' 隐藏"编辑栏"
  ActiveWindow.DisplayWorkbookTabs = False     ' 隐藏"工
'作表标签"
  ActiveWindow.DisplayHeadings = False  ' 隐藏"行号列标"
  On Error Resume Next
' 声明变量
  Dim i As Integer
  Dim NewMenu As CommandBarPopup
  Dim MenuItem As CommandBarControl
' 隐藏常规菜单项
  For m = 1 To Application.CommandBars(1).Controls.Count
    Application.CommandBars(1).Controls(m).Visible = False
  Next
' 删除该菜单
  CommandBars(1).Controls("【基础设置】").Delete
  CommandBars(1).Controls("【教师任课】").Delete
  CommandBars(1).Controls("【进入排课】").Delete
  CommandBars(1).Controls("【查询课表】").Delete
  CommandBars(1).Controls("【关闭系统】").Delete
' 设置新临时菜单 1
  Set NewMenu = CommandBars (1).Controls.Add (Type:=
msoControlPopup, Temporary:=True)
  NewMenu.Caption = "【基础设置】" ' 菜单标题
          Set MenuItem = NewMenu.Controls.Add (Type:=
msoControlButton) ' 菜单项 1
  With MenuItem
    .BeginGroup = True
    .Caption = "基础设置"
    .FaceId = 162
    .OnAction = "基础设置"
  End With
' 设置新临时菜单 1
  Set NewMenu = CommandBars (1).Controls.Add (Type:=
msoControlPopup, Temporary:=True)
  NewMenu.Caption = "【教师任课】" ' 菜单标题
    Set MenuItem = NewMenu.Controls.Add (Type: =
msoControlButton) ' 菜单项 2
  With MenuItem
    .BeginGroup = True
    .Caption = "进入教师任课"
    .FaceId = 39
    .OnAction = "进入处理"
  End With
' 设置新临时菜单 2
  Set NewMenu = CommandBars (1).Controls.Add (Type:=
msoControlPopup, Temporary:=True)
  NewMenu.Caption = "【进入排课】" ' 菜单标题
    Set MenuItem = NewMenu.Controls.Add (Type: =
msoControlButton) ' 菜单项 1
  With MenuItem
    .BeginGroup = True
    .Caption = "进入排课"
    .FaceId = 39
    .OnAction = "进入排课"
  End With
' 设置新临时菜单 3
  Set NewMenu = CommandBars (1).Controls.Add (Type:=
msoControlPopup, Temporary:=True)
  NewMenu.Caption = "【查询课表】" ' 菜单标题
    Set MenuItem = NewMenu.Controls.Add (Type: =
msoControlButton) ' 菜单项 1
  With MenuItem
    .BeginGroup = True
    .Caption = "查询课表"
    .FaceId = 202 '25
    .OnAction = "进入查询"
  End With
' 设置新临时菜单 4
  Set NewMenu = CommandBars (1).Controls.Add (Type:=
msoControlPopup, Temporary:=True)
  NewMenu.Caption = "【关闭系统】" ' 菜单标题
    Set MenuItem = NewMenu.Controls.Add (Type: =
msoControlButton) ' 菜单项 1
```

```
With MenuItem
  .BeginGroup = True
  .Caption = "关闭系统"
  .FaceId = 1088
  .OnAction = "关闭系统"
End With
'设置可浏览区域和显示比例
Sheets(1).Select
  Worksheets(1).ScrollArea = "A1:R43" '可浏览数据区域(固
'定窗口用)
  Range("A1:R43").Select
  ActiveWindow.Zoom = True '显示比例
  If ActiveWindow.Zoom > 100 Then ActiveWindow.Zoom =
100
  Range("C9").Select
  Application.ScreenUpdating = True '恢复屏显
End Sub
Sub 系统界面()
  Application.ScreenUpdating = False '关闭屏显
  On Error Resume Next
  If ActiveSheet.Name = "基础设置" Then Call 基础返回
  If ActiveSheet.Name = "教师任课" Then Call 任课返回
  If Cells(2, 256) = "有错" Then
    Exit Sub
  End If
  Sheets("系统界面").Select
  Range("C9").Select
'声明变量
  Dim m As Integer
'隐藏常规菜单项
  For m = 11 To Application.CommandBars(1).Controls.Count
    Application.CommandBars(1).Controls(m).Visible = True
  Next
  Application.ScreenUpdating = True '恢复屏显
End Sub
```

【模块 1—基础设置】中的代码：

```
Sub 基础设置()
  Application.ScreenUpdating = False '关闭屏显
  On Error Resume Next
'声明变量
  Dim m As Integer
  Dim NewMenu As CommandBarPopup
  Dim MenuItem As CommandBarControl
'隐藏常规菜单项
  For m = 11 To Application.CommandBars(1).Controls.Count
    Application.CommandBars(1).Controls(m).Visible = False
  Next
'设置新菜单为临时的
  Set NewMenu = CommandBars (1).Controls.Add (Type:=
msoControlPopup, Temporary:=True)
  NewMenu.Caption = "【返回】" '菜单标题
    Set MenuItem = NewMenu.Controls.Add (Type: =
msoControlButton) '菜单项
  With MenuItem
    .BeginGroup = True
    .Caption = "返回系统界面"
    .FaceId = 162
    .OnAction = "系统界面"
  End With
  ActiveWindow.DisplayHeadings = False '隐藏"行号列标"
  Application.ScreenUpdating = True '恢复屏显
  Sheets("基础设置").Select
  Range("A2").Select
End Sub
Sub 基础返回()
  Application.ScreenUpdating = False '关闭屏显
  Worksheets(2).Unprotect Password:="zyz2978" '撤销密码
  Cells(2, 256) = ""
'声明变量
  Dim h, i, m, n As Integer
  Dim TJS, ZTS As Integer 'TJS-每天节数 ZTS-每周天数
  Dim JBM As String 'BJM-节编号
'读取数据
  TJS = Cells(2, 4) 'TJS-每天节数
  ZTS = Cells(2, 5) 'ZTS-每周天数
'检查[每天节数]是否含非阿拉伯数字
  For m = 2 To Len(TJS)
    If InStr("0123456789", Mid(TJS, m, 1)) = 0 Then
      Cells(2, 4).Select
      MsgBox "[每天节数]含非阿拉伯数字,请处理。", 48, "【提
示】"
      Cells(2, 256) = "有错"
      Exit Sub
    End If
  Next
'检查[每周天数]是否含非阿拉伯数字
  For m = 2 To Len(ZTS)
    If InStr("0123456789", Mid(ZTS, m, 1)) = 0 Then
      Cells(2, 5).Select
      MsgBox "[每周天数]含非阿拉伯数字,请处理。", 48, "【提
示】"
      Cells(2, 256) = "有错"
      Exit Sub
    End If
  Next
'检查[每周天数]是否超范围
  If ZTS > 7 Then
    Cells(2, 5).Select
    MsgBox "[每周天数]最大值为 7,请处理。", 48, "【提示】"
    Cells(2, 256) = "有错"
    Exit Sub
  End If
'计算最后行号
  h = [A65536].End(xlUp).Row
```

```
If h = 1 Then h = 2
'清除无[班级编号]的班级名称
If h < [B65536].End(xlUp).Row Then
  Range("A" & h + 1 & ":B65536").ClearContents
End If
'检查[班级编号]和[班级名称]中是否有空格
For m = 2 To h
  For n = 1 To 2
    Cells(m, n) = Trim(Cells(m, n))
    Cells(m, n).Select
    If Cells(m, n) = "" Then
      Selection.Interior.ColorIndex = 35
      MsgBox "发现空格,请处理。 ", 48, "【提示】"
      Cells(2, 256) = "有错"
      Exit Sub
    Else
      Selection.Interior.ColorIndex = 2
    End If
  Next
Next
'检查[班级编号]是否为三位阿拉伯数字
For m = 2 To h
  If Len(Cells(m, 1)) <> 3 Then
    Cells(m, 1).Select
    Selection.Interior.ColorIndex = 35
    MsgBox "[班级编号]不是三位,请处理。 ", 48, "【提示】"
    Cells(2, 256) = "有错"
    Exit Sub
  Else
    For n = 1 To 3
      If InStr("0123456789", Mid(Cells(m, 1), n, 1)) = 0 Then
        Cells(m, 1).Select
        Selection.Interior.ColorIndex = 35
        MsgBox "[班级编号]含非阿拉伯数字,请处理。    ", 48,
"【提示】"
        Cells(2, 256) = "有错"
        Exit Sub
      End If
    Next
  End If
Next
'按【班级编号】排序
  Range ("A1:B" & h).Sort Key1: =Range ("A1"), Order1: =
xlAscending, Header:= _
      xlGuess, OrderCustom: =1, MatchCase: =False,
Orientation:=xlTopToBottom, _
                  SortMethod: =xlPinYin, DataOption1: =
xlSortTextAsNumbers
  Range("A2").Select
'检查[班级]数据是否有改动
  For m = 2 To h
    For n = 1 To 2
      If Cells(m, n) <> Cells(m, 250 + n) Then
        YN = MsgBox("是否保存新改动后的[班级]数据? ", 4 +
32 + 0, "【提示】")
        If YN = 6 Then
          '备份[班级]数据、填写【各班课表】工作表 -- 班级编号
        Columns("IQ:IR").ClearContents '清除备份的[班级]数据
          Sheets("各班课表").Range("A2:A65536").ClearContents
'清除【各班课表】中的班级编号
          For i = 2 To h
            Cells(i, 251) = Cells(i, 1)
            Cells(i, 252) = Cells(i, 2)
            Sheets("各班课表").Cells(i, 1) = Cells(i, 1)
          Next
          GoTo A:
        Else
          '恢复备份数据
          Range("A2:B65536").ClearContents '清除变动的[班级]
'数据
          For i = 2 To [IQ65536].End(xlUp).Row
            Cells(i, 1) = Cells(i, 251)
            Cells(i, 2) = Cells(i, 252)
          Next
          GoTo A:
        End If
      End If
    Next
  Next
A:
'检查[每天节数]和[每周天数]是否有改动
  If Cells(2, 4) <> Cells(2, 254) Or Cells(2, 5) <> Cells(2, 255)
Then
    YN = MsgBox("是否保存新[每天节数]或[每周天数]数据?
", 4 + 32 + 0, "【提示】")
    If YN <> 6 Then
      If Cells(2, 4) <> Cells(2, 254) Then Cells(2, 4) = Cells(2,
254)
      If Cells(2, 5) <> Cells(2, 255) Then Cells(2, 5) = Cells(2,
255)
      GoTo B:
    End If
  End If
'备份变动后的[每天节数]或[每周天数]数据
  If Cells(2, 4) <> Cells(2, 254) Then Cells(2, 254) = Cells(2,
4)
  If Cells(2, 5) <> Cells(2, 255) Then Cells(2, 255) = Cells(2,
5)
'清除【教师任课】工作表第一行数据
  Sheets("教师任课").Range("E1:IV1").ClearContents
'填写【教师任课】工作表 -- 可用节数
  For m = 1 To TJS * ZTS
    Sheets("教师任课").Cells(1, 4 + m) = "可用" & m
  Next
```

```
'填写【教师任课】工作表 -- 各节编号
 For n = 1 To ZTS
   For m = 1 To TJS
     JBM = n & Right(100 + m, 2)
     Sheets("教师任课").Cells(1, 6 + TJS * ZTS + (n - 1) *
TJS + m) = JBM
   Next
 Next
'清除【各班课表】工作表第一行数据
 Sheets("各班课表").Range("B1:IV1").ClearContents
'填写【各班课表】工作表 -- 各节编号
 For n = 1 To ZTS
   For m = 1 To TJS
     JBM = n & Right(100 + m, 2)
     Sheets (" 各班课表").Cells (1, 1 +  (n - 1) * TJS + m) =
JBM
   Next
 Next
B:
'填写【系统封面】工作表 -- 单位名称
 If Cells(2, 6) = "" Then
   Sheets("系统界面").Cells(4, 4) = ""
   Sheets("系统界面").Cells(9, 3) = "无终排课系统"
 Else
   Sheets("系统界面").Cells(4, 4) = Cells(2, 6)
   Sheets("系统界面").Cells(9, 3) = "教师任课排课系统"
 End If
 Worksheets(2).Protect Password:="zyz2978"  '密码保护
'删除临时菜单
 CommandBars(1).Controls("【返回】").Delete
 Application.ScreenUpdating = True '恢复屏显
End Sub
```

【模块 2—教师任课】中的代码：

```
Sub 进入处理()
 Application.ScreenUpdating = False '关闭屏显
 On Error Resume Next
'声明变量
 Dim m As Integer
 Dim NewMenu As CommandBarPopup
 Dim MenuItem As CommandBarControl
'隐藏常规菜单项
 For m = 11 To Application.CommandBars(1).Controls.Count
   Application.CommandBars(1).Controls(m).Visible = False
 Next
'设置新菜单为临时的
 Set NewMenu = CommandBars (1).Controls.Add (Type:=
msoControlPopup, Temporary:=True)
 NewMenu.Caption = "【处理教师任课】" '菜单标题
       Set MenuItem = NewMenu.Controls.Add (Type: =
msoControlButton) '菜单项 1
 With MenuItem
   .BeginGroup = True
   .Caption = "录入教师任课"
   .FaceId = 162
   .OnAction = "录入任课"
 End With
       Set MenuItem = NewMenu.Controls.Add (Type:=
msoControlButton) '菜单项 2
 With MenuItem
   .BeginGroup = True
   .Caption = "按姓名修改任课"
   .FaceId = 202
   .OnAction = "修改任课"
 End With
       Set MenuItem = NewMenu.Controls.Add (Type:=
msoControlButton) '菜单项 3
 With MenuItem
   .BeginGroup = True
   .Caption = "删除全部教师任课"
   .FaceId = 232
   .OnAction = "删除全部"
 End With
'设置新菜单为临时的
 Set NewMenu = CommandBars (1).Controls.Add (Type:=
msoControlPopup, Temporary:=True)
 NewMenu.Caption = "【返回】" '菜单标题
       Set MenuItem = NewMenu.Controls.Add(Type:=
msoControlButton) '菜单项
 With MenuItem
   .BeginGroup = True
   .Caption = "返回系统界面"
   .FaceId = 162
   .OnAction = "系统界面"
 End With
 ActiveWindow.DisplayHeadings = False '隐藏"行号列标"
 Application.ScreenUpdating = True '恢复屏显
 Sheets("教师任课").Select
 Range("E2").Select
End Sub
Sub 录入任课()
'声明变量
 Dim h, m As Integer
 Dim XK() As String 'XK-学科
 Dim DD() As String 'DD-地点
'赋值下拉列表框
 h = Sheets("基础设置").[I65536].End(xlUp).Row
 ReDim XK(h, 2) As String
 For m = 2 To h
   XK(m, 1) = Sheets("基础设置").Cells(m, 8)
   XK(m, 2) = Sheets("基础设置").Cells(m, 9)
 Next
 For m = 2 To h
   UserForm1.ComboBox1.AddItem XK(m, 1) & XK(m, 2)
 Next
```

```
h = Sheets("基础设置").[L65536].End(xlUp).Row
ReDim DD(h, 2) As String
For m = 2 To h
  DD(m, 1) = Sheets("基础设置").Cells(m, 11)
  DD(m, 2) = Sheets("基础设置").Cells(m, 12)
Next
For m = 2 To h
  UserForm1.ComboBox2.AddItem DD(m, 1) & DD(m, 2)
  Next
 ' 显示窗体 1
  UserForm1.Show
End Sub
Sub 修改任课()
  Application.ScreenUpdating = False ' 关闭屏显
 ' 声明变量
  Dim h, m As Integer
  Dim JS() As String 'JS-教师
 ' 计算最后行
  h = [A65536].End(xlUp).Row
  If h = 1 Then
    MsgBox "系统中没有任课教师信息。 ", 48, "【提示】"
    Exit Sub
  End If
 ' 调整编号
  For m = 2 To h
    Cells(m, 1) = Right(999 + m, 3)
  Next
 ' 赋值下拉列表框
  ReDim JS(h, 2) As String 'JS-教师
  For m = 2 To h
    JS(m, 1) = Sheets("教师任课").Cells(m, 1)
    JS(m, 2) = Sheets("教师任课").Cells(m, 2)
  Next
  For m = 2 To h
    UserForm2.ComboBox1.AddItem JS(m, 1) & JS(m, 2)
  Next
  Application.ScreenUpdating = True ' 恢复屏显
 ' 显示窗体 2
  UserForm2.Show
End Sub
Sub 删除全部()
  Application.ScreenUpdating = False ' 关闭屏显
  Sheets("教师任课").Select
  YN = MsgBox("是否清除【全部】教师任课信息？ ", 4 + 32 +
256, "【提示】")
  If YN = 6 Then
    YN = MsgBox("确定要清除【全部】教师任课信息吗？ ", 4 +
16 + 256, "【提示】")
    If YN = 6 Then
      Rows("2:65536").ClearContents
      Sheets("各班课表").Rows("2:65536").ClearContents
    End If
  End If
  Application.ScreenUpdating = True ' 恢复屏显
End Sub
Sub 任课返回()
  Application.ScreenUpdating = False ' 关闭屏显
  On Error Resume Next
 ' 删除临时菜单
  CommandBars(1).Controls("【处理教师任课】").Delete
  CommandBars(1).Controls("【返回】").Delete
 'ActiveWorkbook.Save
  Application.ScreenUpdating = True ' 恢复屏显
End Sub
```

【UserForm1】窗体中的代码：

```
Private Sub UserF orm_QueryClose (Cancel As Integer,
CloseMode As Integer) ' 屏蔽【关闭】按钮
  If CloseMode <> 1 Then Cancel = 1
End Sub
Private Sub CommandButton1_Click()
  Application.ScreenUpdating = False ' 关闭屏显
 ' 声明变量
  Dim h, i, l, m As Integer
  Dim BS, JS As Integer
 ' 检查录入数据的合理性
 ' 检查 TextBox1 是否为空
  If TextBox1 = "" Then
    MsgBox "【教师姓名】为空，请核实。 ", 48, "【提示】"
    TextBox1.SetFocus
    Exit Sub
  End If
 ' 检查 TextBox2 是否为三位
  If Len(TextBox2) <> 3 Then
    MsgBox "【班级编号】不是三位，请核实。", 48, "【提示】"
    TextBox2.SetFocus
    Exit Sub
  End If
 ' 检查 TextBox2 是否含非阿拉伯数字
  For m = 1 To Len(TextBox2)
    If InStr("0123456789", Mid(TextBox2, m, 1)) = 0 Then
      MsgBox "【班级编号】 含非阿拉伯数字，请核实。 ", 48, "
【提示】"
      TextBox2.SetFocus
      Exit Sub
    End If
  Next
 ' 检查 TextBox2 班级编号是否存在
  For m = 2 To Sheets("基础设置").[A1].End(xlDown).Row
    If Sheets("基础设置").Cells(m, 1) = TextBox2 Then GoTo
A:
  Next
  MsgBox "【班级编号】 [" & TextBox2 & "] 不存在，请核实。
", 48, "【提示】"
  TextBox2.SetFocus
```

```
  Exit Sub
A:
 ´检查 TextBox3 是否为空
  If TextBox3 = "" Then
    MsgBox "该班【周节数】为空,请核实。", 48, "【提示】"
    TextBox3.SetFocus
    Exit Sub
  End If
 ´检查 TextBox3 是否为两位
  If Len(TextBox3) <> 2 Then
    MsgBox "该班【周节数】不是两位,请核实。 ", 48, "【提示】"
    TextBox3.SetFocus
    Exit Sub
  End If
 ´检查 TextBox3 是否含非阿拉伯数字
  For m = 1 To Len(TextBox3)
    If InStr("0123456789", Mid(TextBox3, m, 1)) = 0 Then
      MsgBox "该班【周节数】含非阿拉伯数字,请核实。
", 48, "【提示】"
      TextBox3.SetFocus
      Exit Sub
    End If
  Next
 ´检查 TextBox3 是否超周总节数
  If Val(TextBox3) > Sheets("基础设置").Cells(2, 4) * Sheets("
基础设置").Cells(2, 5) Then
    MsgBox "该班【周节数】超范围,请核实。 ", 48, "【提示】"
    TextBox3.SetFocus
    Exit Sub
  End If
 ´检查 ComboBox1 是否选择
  If Me.ComboBox1 = "选择学科" Then
    MsgBox "请选择学科。", 48, "【提示】"
    ComboBox1.SetFocus
    Exit Sub
  End If
 ´检查 ComboBox2 是否选择
  If Me.ComboBox1 = "选择上课地点" Then
    MsgBox "请选择上课地点。 ", 48, "【提示】"
    ComboBox2.SetFocus
    Exit Sub
  End If
 ´计算最后行号
  h = [A65536].End(xlUp).Row
 ´录入教师任课信息
  For m = 2 To h
   ´查找教师
    If Cells(m, 2) = TextBox1 Then
     ´计算最后列
      l = Cells(m, 1).End(xlToRight).Column
     ´填写数据 — [班级编号]+[学科编号]+[班周节数]+[上课地
´点编号](共 9 位数)
      For i = 5 To l
        If Left(Cells(m, i), 3) = TextBox2 Then
          Cells (m, i) = TextBox2 & Left (ComboBox1, 2) &
TextBox3 & Left(ComboBox2, 2)
          GoTo B:
        End If
      Next
      Cells (m, l + 1) = TextBox2 & Left (ComboBox1, 2) &
TextBox3 & Left(ComboBox2, 2)
B:
     ´计算最后列
      l = Cells(m, 1).End(xlToRight).Column
     ´计算已录[班数]和[节数]
      BS = l – 4
      Label7.Caption = BS ´[累计任教班数]
      Cells(m, 3) = BS
      JS = 0
      For i = 5 To l
        JS = JS + Val(Mid(Cells(m, i), 6, 2))
      Next
      Label9.Caption = JS ´[累计课时数]
      Cells(m, 4) = JS
      GoTo c:
    End If
  Next
  ´填写数据 — [编号]、[姓名]
  Cells(h + 1, 1) = Right(1000 + h, 3)
  Cells(h + 1, 2) = TextBox1
  ´填写数据 — [班级编号]+[学科编号]+[班周节数]+[上课地点
´编号](共 9 位数)
   Cells (h + 1, 5) = TextBox2 & Left (ComboBox1, 2) &
TextBox3 & Left(ComboBox2, 2)
  ´计算已录[班数]和[节数]
  BS = 1
  Label7.Caption = BS ´[累计任教班数]
  Cells(h + 1, 3) = BS
  JS = Mid(Cells(h + 1, 5), 6, 2)
  Label8.Caption = JS ´[累计课时数]
  Cells(h + 1, 4) = JS
c:
  TextBox2 = ""
  TextBox2.SetFocus
  Application.ScreenUpdating = True ´恢复屏显
End Sub
Private Sub CommandButton2_Click()
  Application.ScreenUpdating = False ´关闭屏显
  TextBox1 = ""
  TextBox2 = ""
  TextBox3 = ""
  TextBox4 = ""
  TextBox1.SetFocus
```

(下转第 40 页)

# 基于注解的 Spring3 MVC 程序开发

郭海伟

**摘 要**：基于注解的 Spring3 MVC 是广泛使用的 MVC 开发方式，结构简单学习难度小、功能强大、灵活性扩展性高。

**关键词**：注解；Spring3 MVC 开发；Controller 包

## 1 引言

从事 JavaEE 软件开发过程中使用其他的一些框架比如 Webwork、Struts2,XML 文件的配置工作量都比较大，随着工程逻辑复杂度的上升，XML 文件也越来越难以维护，而使用 Spring3 MVC 可以基于全注解驱动，代码编写极为方便，模块代码耦合度低。

## 2 实现过程

首先在 myeclipse10.0 工程中新建 Web project（如图 1 所示），工程自动在 WebRoot 目录下生成一个 index.jsp 文件，拷贝 Spring 核心 jar 包和 Spring web jar 包到 WEB-INF\lib 目录下，如图 2 所示。使用 Spring MVC，要配置 DispatcherServle 用于拦截匹配的请求。DispatcherServlet 就是一个 Servlet，所以可以配置多个 DispatcherServlet。DispatcherServlet 是前置控制器，配置在 web.xml 文件中的。

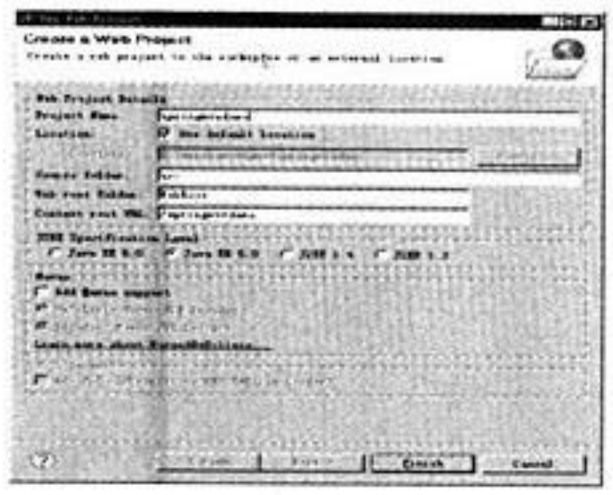


图 1 新建项目

图 2 Lib 目录显示

（1）配置 web.xml 文件

```
<! -- 配置 DispatcherServlet-->
<display-name>SpringMVC</display-name>
 <servlet>
    <servlet-name>spring</servlet-name> <servlet-class>
org.springframework.web.servlet.DispatcherServlet
</servlet-class>
<! -- 可以自定义 servlet.xml 配置文件的位置和名称,默认为
WEB-INF 目录下，名称为 [<servlet-name>]-servlet.xml,如
spring-servlet.xml--! >
 <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
      <servlet-name>spring</servlet-name>
          <url-pattern>*.do</url-pattern>
     </servlet-mapping>
  <welcome-file-list>
   <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

（2）在 WEB-INF 路径下添加 spring-servlet.xml

内容如下：

```
<! -- 设置使用注解的类所在的 jar 包 -->
  <context:component-scan base-package="controller" />
<bean  id = "viewResolver"  class = "org.springframework.web.
servlet.view.UrlBasedViewResolver">
<!-- 对模型视图名称的解析,在请求时模型视图名称添加前后
缀 -->
<property  name = "viewClass"  value = "org.springframework.
web.servlet.view.JstlView" />
<property name="prefix" value="/WEB-INF/view/" />
<property name="suffix" value=".jsp" />
</bean>
```

（3）创建 controller

在 src 目录下新建 HelloWorldController.java 包路径为 controller。

创建一个 Controller 时，不需要继承任何类或实现接口，只需要加入 @Controller 的注解即可。将类名前加上该注解，

当 Spring 启动或者 Web 服务启动 Spring 会自动扫描所有包：

```
package controller;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMap
ping;
import org.springframework.web.servlet.ModelAndView;
@Controller
public class HelloWorldController {
    @RequestMapping("/hello")
    public ModelAndView helloWorld() {
    String message = "Hello World, 基于注解的 Spring
MVC! ";
    System.out.println(message);
      return new ModelAndView ("hello", "message",
message);
    }
}
```

真正让 HelloWorldController 具备 Spring MVC Controller 功能的是 @RequestMapping 这个注解。@RequestMapping 可以标注在类定义处，将 Controller 和特定请求关联起来。

(4) 修改工程自动创建的 WebContent 目录下的 index.jsp 文件，添加一个链接

```
<body>
 <a href="hello.do">Say Hello</a>
</body>
```

(5) WebRoot 目录下新建 view 文件夹，同时在 view 文件夹下新建 hello.jsp 文件来显示问候语：

```
<body>
  Hello World, 基于注解的 Spring MVC!  <br>
</body>
```

## 3 结语

开发完成，启动 tomcat，测试访问，如图 3 所示。

图 3 测试效果：


图 3 测试效果

单击 SayHello 链接，进入 Controller 处理后返回 hello.jsp 页面显示问候语，同时在控制台输入问候语，如图 4，图 5 所示。

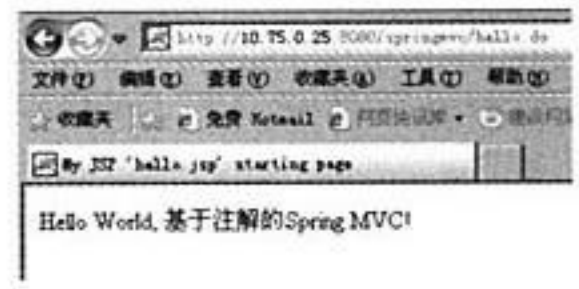



图 4 hello.jsp 页测试

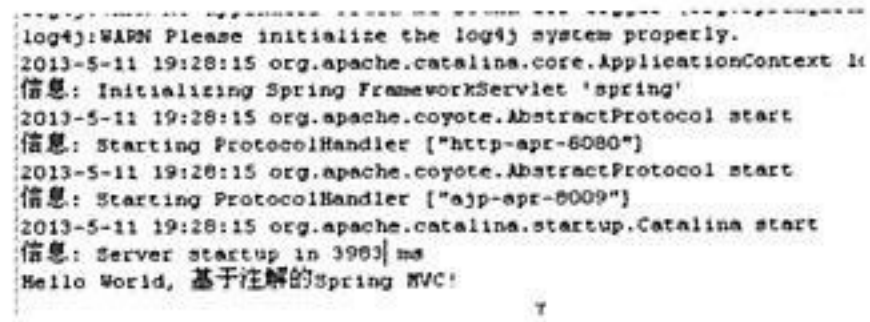



图 5 控制台输入测试

## 参考文献

[1] Seth Ladd, Darren Davison, 等. Expert Spring MVC and Web Flow . Apress, 2008.

[2] 陈雄华，林开雄. Spring 3.x 企业应用开发实战. 电子工业出版社，2012.

[3] Craig Walls , Ryan Breidenbach. Spring in Action. Manning Publications , 2008.

[4] 李刚. 轻量级 Java EE 企业应用实战. 电子工业出版社，2012.

(收稿日期：2013-05-20)

(上接第 13 页)

种查询情况，无论按城市模糊查询站点，还是按站点精确查询，本程序都可自适应得到相应的查询结果。

## 5 结语

利用 SQLite、Jena、IKVM.NET、百度地图 API，笔者完成了具有语义推理功能的火车交通查询程序，既可以模糊查询又可以精确查询。其中，Jena 的本体操作是程序的核心。

通过程序的运行情况可以看出，语义网技术作为地理信息语义查询的一个实现手段应当是可行的。故此，笔者认为在实际运行中，集成 Web3.0 的语义网功能，能够适当提高 WebGIS 的查询性能。目前，这个程序只是初浅地实现了地理信息本体的语义网应用，以后可以进一步深入到有关地理信息语义网方面，加强对于地理本体论的研究。

## 参考文献

[1] 董志. jQuery 集成多源地图服务进行线缓冲区分析[J] . 电脑编程技巧与维护, 2012 (19): 58-66.

[2] http://msdn.microsoft.com/zh-cn/library/w0x726c2 (v=vs.100) . aspx.

[3] 刘甫迎，刘光会，王蓉. C# 程序设计教程 . 2 版. 北京：电子工业出版社，2008.

(收稿日期：2013-01-24)

# 随机序列的几个生成算法与分布模型

明廷堂

**摘　要**：随机序列是科学工程中的一个基本而重要的研究领域，广泛应用于数据结构、算法分析与设计、仿真建模等。对于伪随机序列，研究了几个生成算法及一系列分布模型，并采用 C# 编码完整地刻画了这些算法和模型的行为与特性。为了对它们进行性能分析，编写了详细的可视化测试程序。

**关键词**：随机数；随机序列；随机分布；性能测试

## 1　简介

在数据结构、算法分析与设计、科学模拟等方面都需要用到随机数。由于在数学上，整数是离散型的，实数是连续型的，而在某一具体的工程技术应用中，可能还有数据值的范围性和是否可重复性的要求。

随机数的生成一般有两种方式：一种是硬件方式，另一种是软件方式。一般地，硬件方式生成的随机数质量要好于软件方式生成的随机数。在密码学中，一个随机序列的定义如下：（1）它是随机的；（2）该序列是不可预测的；（3）该序列不能重复产生。随机数生成器有真伪之分，真随机数生成器能满足以上 3 点要求，伪随机数只能满足前两点要求。伪随机数一般用软件方法实现，其基本过程如下：（1）确定一个数学模型或者算法；（2）设置一些参数的值；（3）按照规定的步骤和算法来生成第一个随机数；（4）在第一个随机数的基础上，来生成第二个随机数。重复同样的步骤，从而得到一个随机数序列。

下面首先研究了几个伪随机数生成算法，然后基于这些算法构建几个伪随机数产生器及一系列随机分布模型，最后编写了相关的测试程序。

## 2　随机序列的生成算法及其实现

该模块首先设计了一个抽象的随机序列生成基类 Generator，所有具体的生成算法都从该类继承。Generator 类型声明了所列出的所有随机序列生成算法的通用功能。表 1 简要描述了这些抽象成员。

**表 1　随机序列产生算法抽象基类的成员**

| 成员 | 描述 |
|---|---|
| publicabstractbool CanReset { get;} | 生成器能否重置。该值指示生成器能否产生相同的随机序列。 |
| publicabstractbool Reset () ; | 重置生成器，以便产生相同的随机序列。 |
| publicabstractint Next () ; | 返回一个默认范围的非负随机整数 (0≤x<Int32.MaxValue)。 |
| publicabstractint Next (int maxValue) ; | 返回一个小于指定最大值 maxValue 的非负随机整数 (0≤x<maxValue)。 |
| publicabstractint Next (int minValue, int maxValue) ; | 返回一个小于指定范围的非负随机整数 (minValue≤x<maxValue)。 |
| publicabstractdouble NextDouble () ; | 返回一个默认范围的非负浮点数 (0≤x<1)。 |
| publicabstractdouble NextDouble (double maxValue) ; | 返回一个指定最大值的非负浮点数 (0≤x<maxValue)。 |
| publicabstractdouble NextDouble (double minValue, double maxValue) ; | 返回一个指定范围的非负浮点数 (minValue≤x<maxValue)。 |
| publicabstractbool NextBoolean () ; | 返回一个随机的布尔值。 |
| publicabstractvoid NextBytes (byte [] buffer) ; | 用随机数填充指定的字节数组。每个元素的取值范围为 0≤x≤Byte.MaxValue。 |

这里研究了 4 种伪随机数产生算法。这些算法的具体实现都继承自抽象基类 Generator。下面详细描述这些算法及其编码实现。

标准算法 (StandardGenerator) 是最简单的伪随机数产生器。它在内部使用了一个.NET 框架的 System.Random 类型实例产生伪随机数。下面的代码片段实现了使用该算法产生随机序列的各个接口：

```
publicoverrideint Next()
{
returnthis.generator.Next();
}
publicoverrideint Next(int maxValue)
{
returnthis.generator.Next(maxValue);
}
publicoverrideint Next(int minValue, int maxValue)
{
returnthis.generator.Next(minValue, maxValue);
}
publicoverridedouble NextDouble()
{
returnthis.generator.NextDouble();
}
```

```
publicoverridedouble NextDouble(double maxValue)
{
returnthis.generator.NextDouble() * maxValue;
}
publicoverridedouble NextDouble (double minValue, double
maxValue)
{
double range = maxValue – minValue;
return minValue + this.generator.NextDouble() * range;
}
publicoverridebool NextBoolean()
{
if (this.bitCount == 0)
   {
this.bitBuffer = this.generator.Next();
this.bitCount = 30;
return (this.bitBuffer & 0x1) == 1;
   }
this.bitCount--;
return ((this.bitBuffer >>= 1) & 0x1) == 1;
}
publicoverridevoid NextBytes(byte[] buffer)
{
this.generator.NextBytes(buffer);
}
```

添加剂滞后斐波那契算法 (ALFGenerator) 采用模数产生随机序列。基于斐波那契序列的 ALG 方法是标准的线性同余随机产生器的一种改进。斐波那契序列可以通过下列递归关系产生：$S_n \equiv S_{n-j} * S_{n-k} (\mathrm{mod}\ m), 0 < j < k$。其中：$m$ 是 2 的幂数 $m=2^M$ (通常 $M$=32 或 $M$=64)；操作符 * 表示一般的二进制操作(如加、减、乘、按比特异或等)。这种随机理论非常复杂，其复杂度主要表现在 $j$ 和 $k$ 的选择上。下面的代码片段实现了使用该算法产生随机序列的各个接口：

```
publicoverrideint Next()
{
if (this.i >= this.longLag)  this.Fill();
uint x = this.x[this.i++];
int result = (int)(x >> 1);
if (result == Int32.MaxValue)returnthis.Next();
elsereturn result;
}
publicoverrideint Next(int maxValue)
{
if (this.i >= this.longLag)this.Fill();
uint x = this.x[this.i++];
return    (int)  ((double)  (int)  (x  >>  1)  *  ALFGenerator.
IntToDoubleMultiplier * (double)maxValue);
}
publicoverrideint Next(int minValue, int maxValue)
{
if (this.i >= this.longLag)this.Fill();
uint x = this.x[this.i++];
int range = maxValue – minValue;
if (range < 0)
   {
return minValue + (int)
          ((double)x * ALFGenerator.UIntToDoubleMultiplier *
          ((double)maxValue – (double)minValue));
   }
else
   {
return minValue + (int)
                ((double)(int)(x >> 1) * ALFGenerator.
IntToDoubleMultiplier * (double)range);
   }
}
publicoverridedouble NextDouble()
{
if (this.i >= this.longLag)this.Fill();
uint x = this.x[this.i++];
                return (double)(int)(x >> 1) * ALFGenerator.
IntToDoubleMultiplier;
}
publicoverridedouble NextDouble(double maxValue)
{
if (this.i >= this.longLag)this.Fill();
uint x = this.x[this.i++];
                return   (double) (int) (x >> 1) * ALFGenerator.
IntToDoubleMultiplier * maxValue;
}
publicoverridedouble NextDouble (double minValue, double
maxValue)
{
if (this.i >= this.longLag)this.Fill();
     uint x = this.x[this.i++];
return minValue +   (double) (int) (x >> 1) * ALFGenerator.
IntToDoubleMultiplier * range;
}
publicoverridebool NextBoolean()
{
if (this.bitCount == 0)
   {
if (this.i >= this.longLag)this.Fill();
this.bitBuffer = this.x[this.i++];
this.bitCount = 31;
return (this.bitBuffer & 0x1) == 1;
   }
this.bitCount--;
return ((this.bitBuffer >>= 1) & 0x1) == 1;
}
publicoverridevoid NextBytes(byte[] buffer)
{
```

```
int i = 0; uint w;
while (i < buffer.Length - 3)
   {
if (this.i >= this.longLag)this.Fill();
       w = this.x[this.i++];
       buffer [i++] =  (byte)w; buffer [i++] =  (byte)(w >> 8);
buffer[i++] = (byte)(w >> 16);
       buffer[i++] = (byte)(w >> 24);
   }
if (i < buffer.Length)
   {
if (this.i >= this.longLag)this.Fill();
       w = this.x[this.i++];
       buffer[i++] = (byte)w;
if (i < buffer.Length)
       {
           buffer[i++] = (byte)(w >> 8);
if (i < buffer.Length)
           {
               buffer[i++] = (byte)(w >> 16);
if (i < buffer.Length)buffer[i] = (byte)(w >> 24);
           }
       }
   }
}
```

梅森旋转算法（MT19937Generator）是一个伪随机数产生算法。它基于一个有限二进制字段上的矩阵线性递归关系，可以快速产生高质量的伪随机数，修正了古典随机数发生算法的很多缺陷。该算法的周期取自梅森素数，通常使用两个相近的变体，不同之处在于使用了不同的梅森素数。一个常用的周期是 MT19937-32 (32 位字长)。还有一个变种是 64 位版本的 MT19937-64。对于一个 $k$ 位的长度，Mersenne Twister 算法会在 $[0,2^k-1]$ 的区间内生成离散型均匀分布的随机数。下面的代码片段实现了使用该算法产生随机序列的各个接口：

```
publicoverrideint Next()
{
if  (this.mti >= MT19937Generator.N) this.GenerateNUInts();

uint y = this.mt[this.mti++];
   y ^= (y >> 11); y ^= (y << 7) & 0x9d2c5680U;
   y ^= (y << 15) & 0xefc60000U; y ^= (y >> 18);
int result = (int)(y >> 1);
if (result == Int32.MaxValue) returnthis.Next();
elsereturn result;
}
publicoverrideint Next(int maxValue)
{
if (this.mti >= MT19937Generator.N)this.GenerateNUInts();
uint y = this.mt[this.mti++];
   y ^= (y >> 11); y ^= (y << 7) & 0x9d2c5680U;
   y ^= (y << 15) & 0xefc60000U; y ^= (y >> 18);
     return  (int)((double)(int)(y >> 1) * MT19937Generator.
IntToDoubleMultiplier *
                        (double)maxValue);
}
publicoverrideint Next(int minValue, int maxValue)
{
if (this.mti >= MT19937Generator.N) this.GenerateNUInts();
 uint y = this.mt[this.mti++];
   y ^= (y >> 11); y ^= (y << 7) & 0x9d2c5680U;
   y ^= (y << 15) & 0xefc60000U; y ^= (y >> 18);
int range = maxValue - minValue;
if (range < 0)
   {
return minValue + (int)
  ((double)y * MT19937Generator.UIntToDoubleMultiplier *
        ((double)maxValue - (double)minValue));
   }
else
   {
return minValue + (int)((double)(int)(y >> 1) *
  MT19937Generator.IntToDoubleMultiplier * (double)range);
   }
}
publicoverridedouble NextDouble()
{
if (this.mti >= MT19937Generator.N)this.GenerateNUInts();
uint y = this.mt[this.mti++];
   y ^= (y >> 11); y ^= (y << 7) & 0x9d2c5680U;
   y ^= (y << 15) & 0xefc60000U; y ^= (y >> 18);
          return (double)(int)(y >> 1) * MT19937Generator.
IntToDoubleMultiplier;
}
publicoverridedouble NextDouble(double maxValue)
{
if (this.mti >= MT19937Generator.N)this.GenerateNUInts();
uint y = this.mt[this.mti++];
   y ^= (y >> 11); y ^= (y << 7) & 0x9d2c5680U;
   y ^= (y << 15) & 0xefc60000U; y ^= (y >> 18);
        return  (double) (int) (y >> 1) * MT19937Generator.
IntToDoubleMultiplier * maxValue;
}
publicoverridedouble NextDouble (double minValue, double
maxValue)
{
double range = maxValue - minValue;
if (this.mti >= MT19937Generator.N)this.GenerateNUInts();
uint y = this.mt[this.mti++];
   y ^= (y >> 11); y ^= (y << 7) & 0x9d2c5680U;
   y ^= (y << 15) & 0xefc60000U; y ^= (y >> 18);
return minValue + (double)(int)(y >> 1) * MT19937Generator.
IntToDoubleMultiplier * range;
```

```
}
publicoverridebool NextBoolean()
{
if (this.bitCount == 32)
   {
if (this.mti >= MT19937Generator.N) this.GenerateNUInts();
uint y = this.mt[this.mti++];
      y ^=  (y >> 11); y ^=  (y << 7) & 0x9d2c5680U; y ^= (y
<< 15) & 0xefc60000U;
this.bitBuffer = (y ^ (y >> 18));
this.bitCount = 1;
return (this.bitBuffer & 0x1) == 1;
   }
this.bitCount++;
return ((this.bitBuffer >>= 1) & 0x1) == 1;
}
publicoverridevoid NextBytes(byte[] buffer)
{
int i = 0; uint y;
while (i < buffer.Length - 3)
   {
if (this.mti >= MT19937Generator.N) this.GenerateNUInts();
      y = this.mt[this.mti++];
      y ^= (y >> 11); y ^= (y << 7) & 0x9d2c5680U;
      y ^= (y << 15) & 0xefc60000U; y ^= (y >> 18);
      buffer[i++] = (byte)y; buffer[i++] = (byte)(y >> 8);
  buffer[i++] = (byte)(y >> 16); buffer[i++] = (byte)(y >> 24);
   }
if (i < buffer.Length)
   {
if (this.mti >= MT19937Generator.N) this.GenerateNUInts();
      y = this.mt[this.mti++];
      y ^= (y >> 11); y ^= (y << 7) & 0x9d2c5680U;
      y ^= (y << 15) & 0xefc60000U; y ^= (y >> 18);
      buffer[i++] = (byte)y;
if (i < buffer.Length)
      {
         buffer[i++] = (byte)(y >> 8);
if (i < buffer.Length)
         {
            buffer[i++] = (byte)(y >> 16);
if (i < buffer.Length)buffer[i] = (byte)(y >> 24);
         }
      }
   }
}
```

异或漂移算法 (XorShift128Generator) 是一种周期为 $2^{128}-1$ 的伪随机数产生方法。与标准方法相比，它的运行速度要快得多。它的计算原理很简单：在 $2^{128}-1$ 的周期内，先按比特位进行异或运算，然后进行向左向右移位运算。下面的代码片段实现了使用该算法产生随机序列的各个接口：

```
publicoverrideint Next()
{
uint t = (this.x ^ (this.x << 11)); this.x = this.y; this.y = this.z;
this.z = this.w;
uint w = (this.w = (this.w ^ (this.w >> 19)) ^ (t ^ (t >> 8)));
int result = (int)(w >> 1);
if (result == Int32.MaxValue) returnthis.Next();
elsereturn result;
}
publicoverrideint Next(int maxValue)
{
uint t = (this.x ^ (this.x << 11)); this.x = this.y; this.y = this.z;
this.z = this.w;
uint w = (this.w = (this.w ^ (this.w >> 19)) ^ (t ^ (t >> 8)));
     return (int)((double)(int)(w >> 1) * XorShift128Generator.
IntToDoubleMultiplier * ?
      (double)maxValue);
}
publicoverrideint Next(int minValue, int maxValue)
{
uint t = (this.x ^ (this.x << 11)); this.x = this.y; this.y = this.z;
this.z = this.w;
uint w = (this.w = (this.w ^ (this.w >> 19)) ^ (t ^ (t >> 8)));
int range = maxValue - minValue;
if (range < 0)
   {
return minValue + (int)
 ((double)w * XorShift128Generator.UIntToDoubleMultiplier *
          ((double)maxValue - (double)minValue));
   }
else
   {
return minValue + (int)((double)(int)(w >> 1) *
XorShift128Generator.IntToDoubleMultiplier * (double)range);
   }
}
publicoverridedouble NextDouble()
{
uint t = (this.x ^ (this.x << 11));
this.x = this.y; this.y = this.z; this.z = this.w;
uint w = (this.w = (this.w ^ (this.w >> 19)) ^ (t ^ (t >> 8)));
          return (double)(int)(w >> 1) * XorShift128Generator.
IntToDoubleMultiplier;
}
publicoverridedouble NextDouble(double maxValue)
{
uint t = (this.x ^ (this.x << 11)); this.x = this.y; this.y = this.z;
this.z = this.w;
uint w = (this.w = (this.w ^ (this.w >> 19)) ^ (t ^ (t >> 8)));
          return   (double)(int)(w >> 1) * XorShift128Generator.
IntToDoubleMultiplier * maxValue;
}
```

```
publicoverridedouble NextDouble (double minValue, double
maxValue)
{
double range = maxValue – minValue;
uint t = (this.x ^ (this.x << 11)); this.x = this.y; this.y = this.z;
this.z = this.w;
uint w = (this.w = (this.w ^ (this.w >> 19)) ^ (t ^ (t >> 8)));
                return minValue +(double) (int) (w >> 1) *
XorShift128Generator.IntToDoubleMultiplier * range;
}
publicoverridebool NextBoolean()
{
if (this.bitCount == 0)
   {
uint t = (this.x ^ (this.x << 11)); this.x = this.y; this.y = this.z;
this.z = this.w;
this.bitBuffer = (this.w = (this.w ^ (this.w >> 19)) ^ (t ^ (t >>
8)));
this.bitCount = 31;
return (this.bitBuffer & 0x1) == 1;
   }
this.bitCount--;
return ((this.bitBuffer >>= 1) & 0x1) == 1;
}
publicoverridevoid NextBytes(byte[] buffer)
{
uint x = this.x;uint y = this.y;uint z = this.z;uint w = this.w;
int i = 0;uint t;
while (i < buffer.Length – 3)
   {
      t = (x ^ (x << 11)); x = y; y = z; z = w;
      w = (w ^ (w >> 19)) ^ (t ^ (t >> 8));
      buffer[i++] = (byte)w; buffer[i++] = (byte)(w >> 8);
 buffer[i++] = (byte)(w >> 16); buffer[i++] = (byte)(w >> 24);
   }
if (i < buffer.Length)
   {
      t = (x ^ (x << 11));x = y;y = z;z = w;
      w = (w ^ (w >> 19)) ^ (t ^ (t >> 8));
      buffer[i++] = (byte)w;
if (i < buffer.Length)
      {
         buffer[i++] = (byte)(w >> 8);
if (i < buffer.Length)
         {
            buffer[i++] = (byte)(w >> 16);
if (i < buffer.Length)buffer[i] = (byte)(w >> 24);
         }
      }
   }
this.x = x;this.y = y;this.z = z;this.w = w;
}
```

## 3 随机序列的分布模型及其实现

该模块首先设计了一个抽象的随机序列分布基类Distribution，所有具体的随机分布类型都从该类继承。Distribution类型声明了文中列出的所有随机序列分布的通用功能。表2简要描述了主要的抽象成员。除了这些抽象成员，Distribution类型还提供了所有随机分布的一些通用实现细节。为了计算随机分布，它还必须与一个随机序列生成器关联。

**表2 随机序列分布模型抽象基类的成员**

| 成员 | 描述 |
|---|---|
| publicabstractdouble Minimum {get; } | 获取随机分布的可能最小值。 |
| publicabstractdouble Maximum {get; } | 获取随机分布的可能最大值。 |
| publicabstractdoubleMean {get; } | 获取随机分布的平均值。 |
| publicabstractdouble Median {get; } | 获取随机分布的中值。 |
| publicabstractdouble Variance {get; } | 获取随机分布的方差。 |
| publicabstractdoubleMode {get; } | 获取随机分布的的模式。 |
| publicabstractdouble NextDouble () ; | 返回一个随机分布的浮点数。 |

这里也研究了一些连续型和离散型随机分布模型。这些模型的具体实现都继承自抽象基类Distribution。下面详细描述9种分布模型及其编码实现。

（1）Beta分布（Beta Distribution）：是一个定义在区间上连续概率分布。它有两个形状参数$\alpha$和$\beta$（$0<\alpha<+\infty$，$0<\beta<+\infty$）作为随机变量$X$（$0\leqslant X\leqslant 1$）的指数。在多个学科研究中，Beta分布用于刻画在有限区间上的随机变量的随机行为。例如，利用Beta分布，可以统计种群遗传学中的等位基因，也可以描述项目管理或控制系统的时间分配等。下面的代码刻画了Beta分布模型：

```
publicoverridedouble Minimum
{
get{ return 0.0; }
}
publicoverridedouble Maximum
{
get { return 1.0; }
}
publicoverridedouble Mean
{
get { returnthis.alpha / (this.alpha + this.beta); }
}
publicoverridedouble Median
{
get { returndouble.NaN; }
}
publicoverridedouble Variance
{
get
   {
return (this.alpha * this.beta) /
```

```
        (Math.Pow(this.alpha + this.beta, 2.0) * (this.alpha +
this.beta + 1.0));
    }
}
publicoverridedouble[] Mode
{
get
    {
if ((this.alpha > 1) && (this.beta > 1))
        {
returnnewdouble[] { (this.alpha – 1.0) / (this.alpha + this.beta
– 2.0) };
        }
elseif ((this.alpha < 1) && (this.beta < 1))
        {
returnnewdouble[] { 0.0, 1.0 };
        }
elseif (((this.alpha < 1) && (this.beta >= 1)) || ((this.alpha ==
1) && (this.beta > 1)))
        {
returnnewdouble[] { 0.0 };
        }
elseif (((this.alpha >= 1) && (this.beta < 1)) || ((this.alpha > 1)
&& (this.beta == 1)))
        {
returnnewdouble[] { 1.0 };
        }
else
        {
returnnewdouble[] { };
        }
    }
}
publicoverridedouble NextDouble()
{
double x = this.gammaDistributionAlpha.NextDouble();
return x / (x + this.gammaDistributionBeta.NextDouble());
}
```

（2）柯西分布（CauchyDistribution）：是一个数学期望、方差均不存在的连续型分布。柯西分布有两个参数 $a$ 和 $\theta$（$-\infty<a<+\infty$，$0<\theta<+\infty$），其概率密度函数的图形为一个钟形，与正态分布类似。柯西分布应用也相当广泛：在物理学上，它作为微分方程的结果描述受迫共振；在光谱学中，它用于描述谱线的形状。下面的代码刻画了柯西分布模型：

```
publicoverridedouble Minimum
{
get {returndouble.MinValue; }
}
publicoverridedouble Maximum
{
get { returndouble.MaxValue; }
}
publicoverridedouble Mean
{
get { returndouble.NaN; }
}
publicoverridedouble Median
{
get{ returnthis.alpha; }
}
publicoverridedouble Variance
{
get{ returndouble.NaN; }
}
publicoverridedouble[] Mode
{
get { returnnewdouble[] { this.alpha }; }
}
publicoverridedouble NextDouble()
{
returnthis.alpha + this.gamma * Math.Tan (Math.PI * (this.
Generator.NextDouble() – 0.5));
}
```

（3）连续均匀分布（ContinuousUniformDistribution）：如果连续型随机变量概率密度函数为 $f(x)=\begin{cases}1/(b-a), a\le x\le b\\0 \quad ,\text{otherwise}\end{cases}$，则称随机变量 $X$ 在区间 $[a,b]$ 上的服从均匀分布，记作 $X\sim U[a,b]$。均匀分布表明随机变量 $X$ 落在 $[a,b]$ 的子区间内的概率只与该子区间长度有关，而与子区间位置无关。因此落在的长度相等的任何子区间内的可能性是相等的，即所谓的均匀性。下面的代码刻画了连续均匀分布模型：

```
publicoverridedouble Minimum
{
get { returnthis.alpha; }
}
publicoverridedouble Maximum
{
get{ returnthis.beta; }
}
publicoverridedouble Mean
{
get {returnthis.alpha / 2.0 + this.beta / 2.0; }
}
publicoverridedouble Median
{
get { returnthis.alpha / 2.0 + this.beta / 2.0; }
}
publicoverridedouble Variance
{
get { returnMath.Pow(this.beta – this.alpha, 2.0) / 12.0; }
}
publicoverridedouble[] Mode
```

```
{
get{ returnnewdouble[] { };}
}
publicoverridedouble NextDouble()
{
returnthis.alpha + this.Generator.NextDouble() * this.helper1;
}
```

（4）厄兰分布（ErlangDistribution）：是一个连续的概率分布，目前广泛应用于通信领域、交通工程、数学生物学等领域的可靠性及排队理论中。它有两个参数：一个正整数形状参量 $k$，一个正实数 $\lambda$。厄兰分布的适用性主要体现在指数分布与 Gamma 分布上：当时 $k$=1，厄兰分布退化成指数分布；当 $k$ 扩展到实数域，它就演化成 Gamma 分布。下面的代码刻画了厄兰分布模型：

```
publicoverridedouble Minimum
{
    get{return 0.0;}
}
publicoverridedouble Maximum
{
    get{returndouble.MaxValue;}
}
publicoverridedouble Mean
{
    get{ returnthis.alpha / this.lambda;}
}
publicoverridedouble Median
{
    get{returndouble.NaN;}
}
publicoverridedouble Variance
{
    get{returnthis.alpha / Math.Pow(this.lambda, 2.0);}
}
publicoverridedouble[] Mode
{
get { returnnewdouble[] { (this.alpha – 1) / this.lambda }; }
}
publicoverridedouble NextDouble()
{
double product = 1.0;
for (int i = 0; i <this.alpha; i++)
  {
    product *= this.Generator.NextDouble();
  }
returnthis.helper1 * Math.Log(1.0 – product);
}
```

（5）指数分布（ExponentialDistribution）：是一种分布函数简单的连续概率分布，可以用来表示独立随机事件发生的时间间隔。例如，旅客进机场的时间间隔、中文维基百科新条目出现的时间间隔等。许多电子产品的寿命分布一般服从指数分布。在可靠性研究中，指数是最常用的一种分布形式，其失效率与时间 t 无关。指数分布可以看作当威布尔分布中的形状系数等于 1 的特殊分布。下面的代码刻画了指数分布模型：

```
publicoverridedouble Minimum
{
    get{  return 0.0;}
}
publicoverridedouble Maximum
{
    get{returndouble.MaxValue;}
}
publicoverridedouble Mean
{
    get{  return 1.0 / this.lambda;}
}
publicoverridedouble Median
{
    get{returnMath.Log(2.0) / this.lambda;}
}
publicoverridedouble Variance
{
    get{  returnMath.Pow(this.lambda, –2.0);}
}
publicoverridedouble[] Mode
{
    get{  returnnewdouble[] {0.0};   }
}
publicoverridedouble NextDouble()
{
        returnthis.helper1 * Math.Log(1.0 – this.Generator.
NextDouble());
}
```

（6）伽玛分布（Gamma DistributionGamma）：一种连续型分布，两个参数分别为：形状参数 $\beta$ 和尺度参数 $a$，当 $\beta$ 为正整数时，分布可看作 $a$ 个独立的指数分布之和，当 $\beta$ 趋向于较大数值时，它近似于正态分布。伽玛分布兼有指数分布和幂分布的特点。下面的代码刻画了伽玛分布模型：

```
publicoverridedouble Minimum
{
    get{  return 0.0;}
}
publicoverridedouble Maximum
{
    get{returndouble.MaxValue;}
}
publicoverridedouble Mean
{
    get{returnthis.alpha * this.theta;}
}
publicoverridedouble Median
{
```

```
        get{  returndouble.NaN;}
}
publicoverridedouble Variance
{
        get{  returnthis.alpha * Math.Pow(this.theta, 2.0);}
}
publicoverridedouble[] Mode
{
        get
        {
if (this.alpha >= 1.0)returnnewdouble[] { (this.alpha – 1.0) *
this.theta };
elsereturnnewdouble[] { };
        }
}
publicoverridedouble NextDouble()
{
        double xi, eta, gen1, gen2;
        do
        {
                gen1 = 1.0 – this.Generator.NextDouble();
                gen2 = 1.0 – this.Generator.NextDouble();
if (gen1 <= this.helper2)
                {
  xi = Math.Pow(gen1 / this.helper2, 1.0 / this.helper1);
                eta = gen2 * Math.Pow(xi, this.helper1 – 1.0);
                }
                else
                {
xi = 1.0 – Math.Log((gen1 – this.helper2) / (1.0 – this.helper2));
                eta = gen2 * Math.Pow(Math.E, –xi);
                }
   } while (eta >Math.Pow(xi, this.helper1 – 1.0) * Math.Pow
(Math.E, –xi));

for (int i = 1; i <= this.alpha; i++)
   {
      xi –= Math.Log(this.Generator.NextDouble());
   }
      return xi * this.theta;
}
```

（7）拉普拉斯分布 (Laplace Distribution)：是一种将两个不同位置的指数分布拼接在一起的连续型分布，亦称为双指数分布。它有两个参数：位置参数 $\mu$ 和尺度参数 $a$。拉普拉斯分布的概率密度函数与正态分布类似，后者用相对于 $\mu$ 平均值的差的平方来表示，而后者用于对 $\mu$ 平均值的差的绝对值来表示。因此，拉普拉斯分布的尾部比正态分布更加平坦。下面的代码刻画了拉普拉斯分布模型：

```
publicoverridedouble Minimum
{
        get{  returndouble.MinValue;}
}
publicoverridedouble Maximum
{
        get{returndouble.MaxValue;}
}
publicoverridedouble Mean
{
        get{  returnthis.mu;}
}
publicoverridedouble Median
{
        get{returnthis.mu;}
}
publicoverridedouble Variance
{
        get{ return 2.0 * Math.Pow(this.alpha, 2.0);      }
}
publicoverridedouble[] Mode
{
get { returnnewdouble[] { this.mu }; }
}
publicoverridedouble NextDouble()
{
double rand = 0.5 – this.Generator.NextDouble();
returnthis.mu – this.alpha * Math.Sign (rand) * Math.Log(2.0
* Math.Abs(rand));
}
```

（8）伯努利分布 (Bernoulli Distribution)：是一种常见的离散型分布。当伯努利试验成功时，随机变量取值 1。若伯努利试验失败，随机变量取值 0。因此，它又称为伯努利分布，又称两点分布。如果随机变量 $X$ 的成功概率为 $p$，则失败概率为 $q$=1–$p$，在 $N$ 次试验后，其成功的期望为 $E$ ($X$) =$p$，方差为 $D$ ($X$) =$p$ (1–$p$)。下面的代码刻画了伯努利分布模型：

```
publicoverridedouble Minimum
{
        get{return 0.0;}
}
publicoverridedouble Maximum
{
        get{  return 1.0;}
}
publicoverridedouble Mean
{
        get{ returnthis.alpha; }
}
publicoverridedouble Median
{
        get{  returndouble.NaN;}
}
publicoverridedouble Variance
{
```

```
        get{ returnthis.alpha * (1.0 – this.alpha);}
}
publicoverridedouble[] Mode
{
        get
        {
if (this.alpha > (1 – this.alpha))returnnewdouble[] { 1.0 };
elseif (this.alpha < (1 – this.alpha))returnnewdouble[] { 0.0 };
elsereturnnewdouble[] { 0.0, 1.0 };
        }
}
publicoverridedouble NextDouble()
{
        if (this.Generator.NextDouble() <this.alpha)return 1.0;
elsereturn 0.0;
}
```

(9) 泊松分布 (Poisson Distribution)：是另一种常见的离散型分布。Poisson 分布是二项分布的一种特殊形式，适用于描述单位时间内随机事件发生的次数的概率。如某一服务设施在一定时间内受到的服务请求的次数、电话交换机接到呼叫的次数、汽车站台的候客人数、机器出现的故障数、自然灾害发生的次数、DNA 序列的变异数、放射性原子核的衰变数等。下面的代码刻画了泊松分布模型：

```
publicoverridedouble Minimum
{
        get{  return 0.0;}
}
publicoverridedouble Maximum
{
        get{  returndouble.MaxValue;}
}
publicoverridedouble Mean
{
        get{  returnthis.lambda;}
}
publicoverridedouble Median
{
        get{  returndouble.NaN;}
}
publicoverridedouble Variance
{
        get{  returnthis.lambda;}
}
publicoverridedouble[] Mode
{
        get
        {
if (this.lambda == Math.Floor(this.lambda))
        {
returnnewdouble[] { this.lambda – 1.0, this.lambda };
        }
else
        {
returnnewdouble[] { Math.Floor(this.lambda) };
        }
        }
}
publicoverridedouble NextDouble()
{
double count = 0.0;
            for (double product = this.Generator.NextDouble();
product >= this.helper1;
            product *= this.Generator.NextDouble())
  {
     count++;
  }
return count;
}
```

本实例还实现了一些其他的连续型和离散型概率分布。限于篇幅，这里不再逐一列出。

## 4 测试程序

为了便于观察随机序列的产生及分布过程，编写了一个可视化的测试程序。

在生成算法的测试项中，详细地比较了各种生成方法的时间性能。图 1 展示了生成算法的测试结果。

**图 1 各种随机序列生成算法的性能比较**

在分布模型的测试项中，主要是分析不同的参数对分布模型的影响。图 2 展示了分布模型的测试结果。

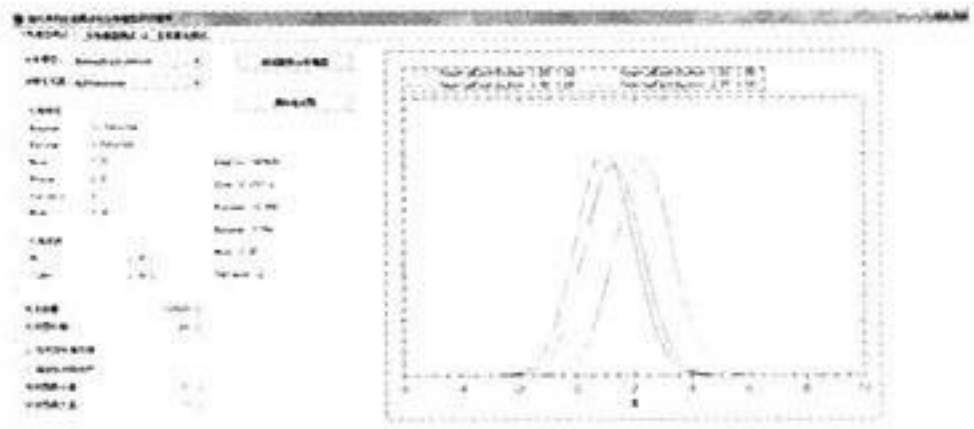


**图 2 不同参数值对随机分布模型的影响**

(收稿日期：2013-04-10)

# VB 实现链表数据结构

刘 烽

**摘 要**：针对 Basic 语言没有指针的缺陷，借鉴面向对象的编程思想，探索了一种利用 Basic 语言实现链表数据结构的方法。在 VB6.0 开发环境下，具体实现了链表的构造、结点的添加和删除等基本操作，程序运行结果表明，Basic 语言也能很好地实现链表这种数据结构。

**关键词**：数据结构；链表；Visual Basic 语言

## 1 问题提出

链表是一种重要的数据结构，广泛应用于多种编程语言中。由于 C 语言具有指针这个基本数据类型，采用 C 语言实现链表是非常直观方便的。然而不具备指针这种数据类型的程序语言，比如 Basic，是不是无法实现链表的这种数据结构呢？答案是否定的，经过探索发现：诸如 Basic 语言等没有指针的高级语言，其实也是可以很好地实现链表这种数据结构的。

## 2 设计思路

链表的种类很多，如单链表、双向链表、循环链表，为简单起见，本文主要以单链表为例，介绍如何在 Visual Basic 6.0 开发环境下实现一个单链表数据结构。单链表的结点结构如图 1 所示。

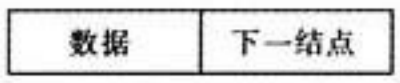



**图 1 单链表的结点**

从图 1 中可以看出：Basic 语言没有指针，似乎无法描述“下一结点”。经过一番思索，笔者联想到：对于面向对象的语言（C++、C#）而言，“类”这个数据类型在内存中的形态其实和“指针”是非常相似的，都是内存的地址。那么，构造一个这样的类：用这个类描述“下一结点”应该就可以实现“指向”的功能了。

## 3 程序实现

在 Visual Basic 6.0 平台下实现单链表需要用到 3 个自定义类：CListNode、CListPointer 和 CList 类。具体代码如下：

### 3.1 CListNode 类

```
'*************************************************
' 文件名:CListNode.cls
' 类名: CListNode
' 成员: mNodeData
'       mNextNode
' 属性:Data
'       NextNode
'*************************************************
Option Explicit
Private mNodeData As Variant     '存储链表结点数据
Private mNextNode As CListNode   '下一个结点
Public Property Get Data() As Variant
    Data = mNodeData
End Property
Public Property Let Data(vNewValue As Variant)
    mNodeData = vNewValue
End Property
Public Property Get NextNode() As CListNode
    Set NextNode = mNextNode
End Property
Public Property Let NextNode( vNewValue As Variant)
    Set mNextNode = vNewValue
End Property
```

### 3.2 CListPointer 类

```
'*************************************************
' 文件名:CListPointer.cls
' 类名: CListPointer
' 成员: mTempNode
'       mFirstNode
' 方法:NextItem
'       HasMoreItems
'       ResetTempNode
'*************************************************
Option Explicit
Private mTempNode As CListNode
Private mFirstNode As CListNode   '指向第一个结点
Public Property Let StartNode( vNewValue As Variant)
    Set mFirstNode = vNewValue
      mTempNode = mFirstNode
End Property
Public Function NextItem() As Variant
Dim tempData As Variant
```

```
    If mTempNode Is Nothing Then
        NextItem = Null
    Else
        tempData = mTempNode.Data
        Set mTempNode = mTempNode.NextNode
        NextItem = tempData
    End If
End Function
Public Function HasMoreItems() As Boolean
    HasMoreItems = IIf (Not mTempNode Is Nothing, True,
False)
End Function
Public Sub ResetTempNode()
    mTempNode = mFirstNode
End Sub
```

### 3.3 CList 类

```
'*****************************************************
' 文件名:CList.cls
' 类名: CList
' 成员: mFirstNode                    //链表首结点
'       mLastNode                     //链表最后结点
' 方法:IsEmpty                        //判断链表是否为空
'       InsertFromFront               //前向插入结结点
'       InsertFromBack                //后向插入结结点
'       RemoveFromFront               //前向删除结结点
'       RemoveFromBack                //后向删除结结点
' 描述: 代表整个单链表
'*****************************************************
Option Explicit On
Private mFirstNode As CListNode
Private mLastNode As ClistNode
Public Function IsEmpty() As Boolean   '判断链表是否为空表
    IsEmpty = IIf(mFirstNode Is Nothing, True, False)
End Function
Public Sub InsertFromFront( insertItem As Variant)
'链表的前向插入操作
    Dim tempNode As CListNode
    If IsEmpty() Then
        mFirstNode = New CListNode
        mLastNode = mFirstNode
    Else
        tempNode = mFirstNode
        mFirstNode = New CListNode
        mFirstNode.NextNode = tempNode
    End If
    mFirstNode.Data = insertItem
End Sub
Public Sub InsertFromBack( insertItem As Variant) '链表的后
'向插入操作
    Dim tempNode As CListNode
    If IsEmpty() Then
        mLastNode = New CListNode
        mFirstNode = mLastNode
    Else
        tempNode = mLastNode
        mLastNode = New CListNode
        tempNode.NextNode = mLastNode
    End If
    mLastNode.Data = insertItem
End Sub
Public Function RemoveFromFront()' 链表的前向删除操作
    Dim removeItem As Variant
    If IsEmpty() Then
        Call MsgBox("链表已经为空! ")
        RemoveFromFront = Null
        Exit Function
    End If
    removeItem = mFirstNode.Data
    If mFirstNode Is mLastNode Then
        mFirstNode = Nothing
        mLastNode = Nothing
    Else
        mFirstNode = mFirstNode.NextNode
    End If
    RemoveFromFront = removeItem
End Function
Public Function RemoveFromBack()' 链表的后向删除操作
    Dim removeItem As Variant
    Dim current As CListNode
    If IsEmpty() Then
        MsgBox("链表已经为空! ")
        RemoveFromBack = Null
        Exit Function
    End If
    removeItem = mLastNode.Data
    If mFirstNode Is mLastNode Then
        mFirstNode = Nothing
        mLastNode = Nothing
    Else
      current = mFirstNode
      While Not current.NextNode Is mLastNode
        current = current.NextNode
      End While
      mLastNode = current
      current.NextNode = Nothing
    End If
    RemoveFromBack = removeItem
End Function
Public Property Get Pointer() As Variant
    Dim clper As CListPointer
    Set clper = New CListPointer
    clper.StartNode = mFirstNode
```

(下转第 63 页)

# A* 寻找最短路径算法及实现

王文举

**摘　要**：介绍 A* 寻找最短路径算法和 C# 代码实现，通过建立完善的网格数据模型，针对网格节点进行计算和维护，使程序流程更加清晰。

**关键词**：C# 语言；A* 算法；寻找最短路径

## 1　引言

A* 算法是静态路网中求解最短路径最有效的一种方法，公式表示为 f (n) =g (n) +h (n)，其中 f (n) 是从初始节点经由节点 n 到目标点的估价函数，g (n) 是从初始节点到 n 节点的实际代价，h (n) 是从 n 到目标节点最佳路径的估计代价。因为 g (n) 是已知的，找到最短路径关键在于估计函数 h (n) 的选取，下文 h (n) 选择为 n 到目标节点的 Hamilton 距离。

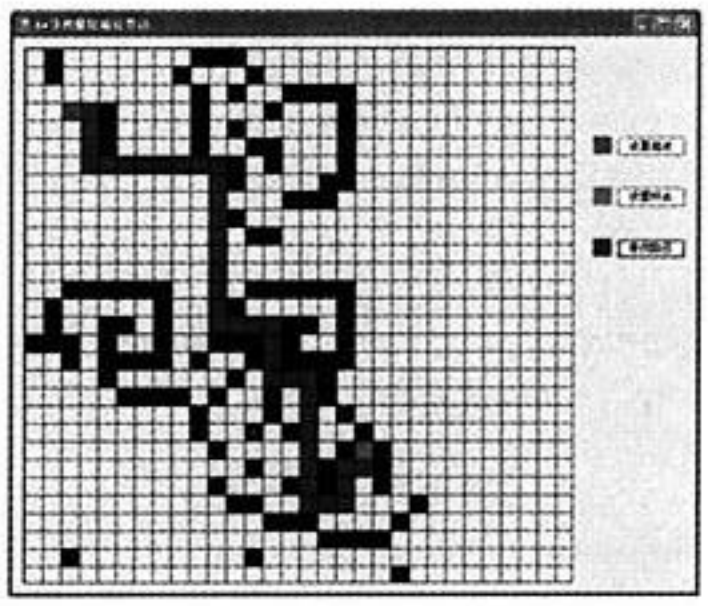


图 1　A* 寻找最短路径算法界面

## 2　设计思路

利用 C# 语言实现 A* 寻找最短路径算法，如图 1 所示，除了窗口界面类外设计了两个类。

一是模型类（MapMode），功能为建立网格数据模型和实现 A* 算法。建立网格数据模型包括：建立网格节点结构模型（Grid），实现模型的构造（MapMode (int [,] _map)），初始化网格节点（InitMapGrid ()）；实现 A* 算法包括：AstarFindPath ( int _startGridId, int _endGridId)，AStarFindPath_step ( int _startGridId, int _endGridId)。

二是画图类（Draw），功能为绘制起点（DrawStaPoint (int x, int y)），绘制终点（DrawEndPoint (int x, int y)），绘制障碍（DrawHinderPoint (int x, int y)），绘制路径（DrawPathPoint (int x, int y)），绘制游戏场景（DrawBackground ()），清空游戏背景（BackgroundClear ()）。

网络资料上，A* 算法的搜索过程一般为创建两个表，OPEN 表保存所有已生成而未考察过的节点，CLOSE 表记录已经访问的节点。以下实现过程没有创建表，而是在网格节点（Grid）中包含字段 open（表示该节点是否在计算列表中），字段 close（表示该节点是否在行走路径中）。通过计算节点相邻的上下左右 4 个节点的 f (n) 的值，并维护节点的字段 open、close 的值，在字段 open 为 true 的所有节点中选择 f (n) 最小的值为下一步格子，实现最短路径搜索。该方法仅针对网格节点进行计算和维护，使程序流程更加清晰。

## 3　实现代码

### 3.1　窗口界面类

```
public partial class Form1 : Form
    {   string btnSelect = "";
        Point startPoint = new Point(-1, -1); //起点
        Point endPoint = new Point(-1, -1); //终点
        Point[] myPath; //路径
        bool isComputing = false;
        Draw myDraw;
        MapMode myMapMode;
          int[,] map = new int[30, 30]  //如图 1 所示,1 表示地形
//消耗,255 表示障碍
{{1,255,1,1,1,1,1,1,1,255,255,255,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1},
{1,255,1,1,1,1,1,1,1,255,1,1,1,255,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1},
{1,1,1,1,1,1,1,1,1,1,255,1,255,1,1,255,255,255,255,1,1,1,1,1,1,1,
1,1,1,1,1,1},
{1,1,1,1,255,1,1,1,1,255,1,1,1,255,1,1,1,255,1,1,1,1,1,1,1,1,1,
1,1,1},
{1,1,1,1,255,1,1,1,1,255,1,255,1,1,1,1,1,255,1,1,1,1,1,1,1,1,1,
1,1,1},
{1,1,1,1,255,1,1,1,1,255,1,1,255,255,1,1,1,255,1,1,1,1,1,1,1,1,
1,1,1,1},
{1,1,1,1,1,1,1,1,1,1,1,1,255,1,255,1,1,1,2
```

```
55,1,1,1,1,1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1,1,1,1,255,1,1,1,1,255,255,1,1,1,1,1,1,1,1,1,1,
1,1},
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,255,255,255,1,1,1,1,1,1,1,1,1,1,
1,1},
{1,1,1,1,1,1,1,1,1,1,1,255,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1,1,1,1,1,255,255,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1},
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
{1,1,255,255,255,255,255,255,1,1,1,1,255,255,255,255,255,2
55,1,1,1,1,1,1,1,1,1,1,1,1},
{1,255,1,1,1,1,1,255,1,1,1,255,1,1,1,1,1,255,1,1,1,1,1,1,1,1,1,
1,1,1},
{1,255,1,1,255,255,1,255,1,1,1,1,1,1,255,255,1,255,1,1,1,1,1,
1,1,1,1,1,1,1},
{255,255,255,1,255,1,1,255,1,1,255,255,255,1,255,1,1,255,1,
1,1,1,1,1,1,1,1,1,1,1,1},
{1,1,255,1,255,255,255,255,1,255,1,1,255,1,255,255,255,255,
1,1,1,1,1,1,1,1,1,1,1,1,1},
{1,1,1,1,255,1,1,1,1,1,1,255,1,1,1,1,255,1,1,1,1,1,1,1,1,1,1,1,
1,1},
{1,1,1,1,1,255,255,255,255,1,255,1,1,255,1,1,255,1,1,1,1,1,1,
1,1,1,1,1,1,1,1},
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
{1,1,255,1,1,1,1,1,1,1,1,1,255,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1},
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,255,1,1,1,1,1,1,1,1,1,1}};
        public Form1()
        {   InitializeComponent(); }
        private void Form1_Load(object sender, EventArgs e)
        {   myDraw = new Draw(panel1);
            myMapMode = new MapMode(map);
        }
        protected override void OnPaint(PaintEventArgs e)
        {   Graphics g = panel1.CreateGraphics();
            myDraw.DrawBackground();
            for (int y = 0; y < myMapMode.height; y++)
                for (int x = 0; x < myMapMode.width; x++)
                {   if (myMapMode.map[y, x] == 255)
                    { myDraw.DrawHinderPoint(x, y); }
                }
            base.OnPaint(e);
        }
        private void btnStaPoint_Click(object sender, EventArgs
e)
    { btnSelect = "StartPoint"; } //按键"设置起点"的响应事件
          private void btnEndPoint_Click (object sender,
EventArgs e)
    { btnSelect = "EndPoint";  } //按键"设置终点"的响应事件
          private void panel1_MouseDown (object sender,
MouseEventArgs e)
        {   if (isComputing)  return;
            if (btnSelect == "")
            { MessageBox.Show("请先选择"设置起点"或"设置
终点"");
                return;
            }
            int x = e.X / MapMode.Node;
            int y = e.Y / MapMode.Node;
            if (btnSelect == "StartPoint")
            {   if (startPoint ! = new Point(-1, -1))
                { myDraw.ErasePoint(startPoint.X, startPoint.Y);
//清空原有的起点
                }
                startPoint = new Point(x, y);//赋值新的起点
                myDraw.DrawStaPoint(x, y);//绘制新的起点
            }
            if (btnSelect == "EndPoint")
            {   if (endPoint ! = new Point(-1, -1))
                {   myDraw.ErasePoint(endPoint.X, endPoint.Y);
//清空原有的终点
                }
                endPoint = new Point(x, y);//赋值新的终点
                myDraw.DrawEndPoint(x, y);//绘制新的终点
            }
        }
        //按键"寻找路径"的响应事件
                private void btnFindPath_Click (object sender,
EventArgs e)
        {   if (isComputing) return;
            if  (startPoint == new Point (-1, -1) || endPoint ==
new Point(-1, -1))
                {   MessageBox.Show (" 请维护好起点和终点");
return;
            }
            isComputing = true;
            if (myPath ! = null && myPath.Length > 0)
            {   for (int j = 0; j < myPath.Length; j++)
```

```
            {   myDraw.ErasePoint(myPath[j].X, myPath[j].Y);
}
            }
            myDraw.DrawStaPoint(startPoint.X, startPoint.Y);
//绘制新的起点
            myDraw.DrawEndPoint(endPoint.X, endPoint.Y);
//绘制新的终点
                        myPath = myMapMode.AStarFindPath
(myMapMode.GetGridID(startPoint.X, startPoint.Y), myMap
Mode.GetGridID(endPoint.X, endPoint.Y));
            if (myPath ! = null && myPath.Length > 0)
            {   for (int i = 1; i < myPath.Length; i++)
                {   myDraw.DrawPathPoint (myPath [i].X, myPath
[i].Y); }
            }
            else
            {   MessageBox.Show("没有连通路径"); return;
            }
            isComputing = false;
        }
    }
```

### 3.2 MapMode 类

```
class MapMode
    {   public struct Grid
        {   public int x;
            public int y;
            public int c;//地形消耗
            public int f;//估价值,估计消耗总值
            public int g;//初始节点到格子的实际代价值,路径消耗
            public int h;//格子到目标节点的估计代价值,预计消耗
            public int GridID;//格子的 ID 编号
            public int upGridID;//上边的格子的 ID
            public int downGridID;//下边的格子的 ID
            public int leftGridID;//左边的格子的 ID
            public int rightGridID;//右边的格子的 ID
            public bool open;//是否在 open 表中,表示是否在计算
//列表中
              public bool close;//是否在 close 表中,表示是否在行
//走路径中
            public int nextGridID;//下一步格子的 ID
            public int prevGridID;//上一步格子的 ID
        public static bool operator ==(Grid a, Grid b)
        {   if (a.x == b.x && a.y == b.y) return true;
            else  return false;
        }
        public static bool operator ! =(Grid a, Grid b)
        {  if (a.x ! = b.x || a.y ! = b.y) return true;
           else  return false;
        }
}
public int[,] map;//地图
        public int height; public int width; //地图的高和宽
        public Grid[] MapGrid;
          public static int Node = 20;//每个格子的宽和高都是 20
//个像素
        public MapMode(int[,] _map)
        {   this.map = _map;
            height = map.GetLength (0);//地图的高
            width = map.GetLength(1);//地图的宽
             MapGrid =new Grid [height * width + 1];//MapGrid
//编号从 1 开始
            InitMapGrid();//对 MapGrid 进行初始化
        }
        public void InitMapGrid()
        {   int id = 1; //MapGrid 编号从 1 开始
            for (int y = 0; y < height; y++)
                for (int x = 0; x < width; x++)
                {   MapGrid[id].x = x;
                    MapGrid[id].y = y;
                    MapGrid[id].c = map[y, x];//地形消耗
                    MapGrid[id].GridID = id;
                     if (x == 0) //第一列,没有左边的格子的 ID,赋值
//为 0
                    { MapGrid[id].leftGridID = 0; }
                    else
                    { MapGrid[id].leftGridID = id – 1; }
                     if (x == width – 1)//最后一列,没有右边的格子
//的 ID
                    { MapGrid[id].rightGridID = 0; }
                    else
                    { MapGrid[id].rightGridID = id + 1; }
                    if (y == 0)//第一行,没有上边的格子的 ID
                    { MapGrid[id].upGridID = 0; }
                    else
                    { MapGrid[id].upGridID = id – width; }
                    if (y == height – 1)//最后一行
                    { MapGrid[id].downGridID = 0; }
                    else
                    { MapGrid[id].downGridID = id + width; }
                    MapGrid[id].open = false;//不在 open 表中
                    MapGrid[id].close = false;//不在 close 表中
                    MapGrid[id].prevGridID = 0;//上一步
                    MapGrid[id].nextGridID = 0;//下一步
                    id++;
                }
        }
        //由 x,y 获得格子的 ID
        public int GetGridID(int x, int y)
        { return y * height + x + 1; }
        //计算 h(n),Hamilton 距离
        public void SetH(int endGridId)
        {   for (int i = 1; i < MapGrid.Length; i++)
            {  MapGrid[i].h = Math.Abs(MapGrid[i].x – MapGrid
```

```
[endGridId].x) + Math.Abs (MapGrid [i].y - MapGrid
[endGridId].y);
        }
    }
    //追溯上一步格子,设置其 nextGridID 的值
    public void FindPrev(int _endGridId)
    {  if (_endGridId == 0) return;
       if (MapGrid[_endGridId].prevGridID ! = 0)
       {  int preGridId = MapGrid[_endGridId].prevGridID;
          if (preGridId == 0) return;
          MapGrid[preGridId].nextGridID = _endGridId;
          FindPrev(preGridId);
       }
    }
        public Point [] AStarFindPath (int _startGridId, int
_endGridId)
    {  //对 MapGrid 的 open 表和 close 表以及上一步、下一
//步进行初始化
       for (int id = 1; id < MapGrid.Length; id++)
       {  MapGrid[id].open = false;//不在 open 表中
          MapGrid[id].close = false;//不在 close 表中
          MapGrid[id].prevGridID = 0;//上一步
          MapGrid[id].nextGridID = 0;//下一步
       }
       SetH(_endGridId);//预计消耗
       List<Point> myPointList=new List<Point>();
        MapGrid[_startGridId].open = true;//起始格子加入计
//算列表
       MapGrid[_startGridId].g = MapGrid[_startGridId].c;
//起始格子路径消耗等于地形消耗
       MapGrid[_startGridId].f = MapGrid[_startGridId].h +
MapGrid[_startGridId].g;
       AStarFindPath_step(_startGridId, _endGridId);
       Grid startGrid = MapGrid[_startGridId];
       while (startGrid.nextGridID ! = 0)
    {  myPointList.Add(new Point(startGrid.x, startGrid.y));
          startGrid = MapGrid[startGrid.nextGridID];
       }
       Point[] myPoint = new Point[myPointList.Count];
       for (int i = 0; i < myPointList.Count; i++)
       {  myPoint[i] = myPointList[i];
       }
       return myPoint;
    }
     private void AStarFindPath_step (int _startGridId, int
_endGridId)
    {  if (_startGridId == _endGridId ) return;
    MapGrid[_startGridId].open = false;//从计算列表中移除
    MapGrid[_startGridId].close = true;//从行走路径中添加
        //上边的格子没有在行走路径中,计算它的路径消耗 g
//和估计消耗总值 h
    if  (MapGrid [_startGridId].upGridID ! = 0 && ! MapGrid
[MapGrid[_startGridId].upGridID].close)
        {  if (MapGrid[MapGrid[_startGridId].upGridID].open)
//上边的格子在计算列表中
           {
              //格子的路径消耗+上边格子的地形消耗<上边格
//子的路径消耗,当前路径比原来路径更近
if (MapGrid[_startGridId].g + MapGrid[MapGrid[_startGridId].
upGridID].c <
MapGrid[MapGrid[_startGridId].upGridID].g)
  { MapGrid [MapGrid [_startGridId].upGridID].prevGridID =
_startGridId;
              MapGrid [MapGrid[_startGridId].upGridID].g
= MapGrid[_startGridId].g + MapGrid[MapGrid[_startGridId].
upGridID].c;//路径消耗
    MapGrid [MapGrid [_startGridId].upGridID].f = MapGrid
[MapGrid [_startGridId].upGridID].g + MapGrid [MapGrid
[_startGridId].upGridID].h; //估计消耗总值
           }
        }
        else //上边的格子不在计算列表中
        {  MapGrid[MapGrid[_startGridId].upGridID].open
=true ;//加入计算列表中
           MapGrid [MapGrid [_startGridId].upGridID].
prevGridID = _startGridId;
           MapGrid [MapGrid [_startGridId].upGridID].g =
MapGrid [_startGridId].g + MapGrid [MapGrid[_startGridId].
upGridID].c;//路径消耗
    MapGrid [MapGrid [_startGridId].upGridID].f = MapGrid
[MapGrid [_startGridId].upGridID].g + MapGrid [MapGrid
[_startGridId].upGridID].h; //估计消耗总值
        }
     }
     //下边的格子没有在行走路径中,计算它的路径消耗 g
//和估计消耗总值 h
if (MapGrid [_startGridId].downGridID ! = 0 && ! MapGrid
[MapGrid[_startGridId].downGridID].close)
     {  if (MapGrid[MapGrid[_startGridId].downGridID].
open)//下边的格子在计算列表中
        { if (MapGrid [_startGridId].g + MapGrid
[MapGrid [_startGridId].downGridID].c < MapGrid [MapGrid
[_startGridId].downGridID].g) //当前路径比原来路径更近
           { MapGrid[MapGrid[_startGridId].downGridID].
prevGridID = _startGridId;
              MapGrid[MapGrid[_startGridId].downGridID].
g = MapGrid[_startGridId].g + MapGrid[MapGrid[_startGridId].
downGridID].c;//路径消耗
    MapGrid[MapGrid[_startGridId].downGridID].f = MapGrid
[MapGrid [_startGridId].downGridID].g + MapGrid [MapGrid
[_startGridId].downGridID].h; //估计消耗总值
           }
        }
        else //下边的格子不在计算列表中
```

```
            {  MapGrid [MapGrid[_startGridId].downGridID].
open = true;//加入计算列表中
                MapGrid  [MapGrid [_startGridId].downGridID].
prevGridID = _startGridId;
               MapGrid[MapGrid[_startGridId].downGridID].g =
MapGrid [_startGridId].g + MapGrid [MapGrid [_startGridId].
downGridID].c;//路径消耗
  MapGrid [MapGrid [_startGridId].downGridID].f = MapGrid
[MapGrid [_startGridId].downGridID].g + MapGrid [MapGrid
[_startGridId].downGridID].h; //估计消耗总值
            }
         }
         //左边的格子没有在行走路径中,计算它的路径消耗 g
//和估计消耗总值 h
 if (MapGrid [_startGridId].leftGridID ! = 0 && ! MapGrid
[MapGrid[_startGridId].leftGridID].close)
            {  if (MapGrid [MapGrid [_startGridId].leftGridID].
open)//左边的格子在计算列表中
               {  if (MapGrid[_startGridId].g + MapGrid[MapGrid
[_startGridId].leftGridID].c < MapGrid [MapGrid [_startGridId].
leftGridID].g)  //当前路径比原来路径更近
                  {  MapGrid [MapGrid [_startGridId].leftGridID].
prevGridID = _startGridId;
                     MapGrid[MapGrid[_startGridId].leftGridID].g
= MapGrid [_startGridId].g + MapGrid [MapGrid [_startGridId].
leftGridID].c;
                     MapGrid [MapGrid [_startGridId].leftGridID].f
= MapGrid [MapGrid [_startGridId].leftGridID].g + MapGrid
[MapGrid[_startGridId].leftGridID].h;
                  }
               }
               else //左边的格子不在计算列表中
               {  MapGrid[MapGrid[_startGridId].leftGridID].open
= true;//加入计算列表中
                  MapGrid  [MapGrid [_startGridId].leftGridID].
prevGridID = _startGridId; MapGrid  [MapGrid [_startGridId].
leftGridID].g = MapGrid[_startGridId].g +
                 MapGrid[MapGrid[_startGridId].leftGridID].c;
                  MapGrid [MapGrid [_startGridId].leftGridID].f =
MapGrid [MapGrid [_startGridId].leftGridID].g + MapGrid
[MapGrid[_startGridId].leftGridID].h;
               }
            }
            //右边的格子没有在行走路径中,计算它的路径消耗 g
//和估计消耗总值 h
 if (MapGrid [_startGridId].rightGridID ! = 0 && ! MapGrid
[MapGrid[_startGridId].rightGridID].close)
            {  if (MapGrid [MapGrid[_startGridId].rightGridID].
open)//右边的格子在计算列表中
               {  if (MapGrid [_startGridId].g + apGrid [MapGrid
[_startGridId].rightGridID].c
< MapGrid[MapGrid[_startGridId].rightGridID].g) //当前路径比
//原来路径更近
                  {  MapGrid [MapGrid [_startGridId].rightGridID].
prevGridID = _startGridId;
                     MapGrid[MapGrid[_startGridId].rightGridID].g
= MapGrid [_startGridId].g + MapGrid [MapGrid [_startGridId].
rightGridID].c;
                     MapGrid [MapGrid [_startGridId].rightGridID].f
= MapGrid [MapGrid [_startGridId].rightGridID].g + MapGrid
[MapGrid[_startGridId].rightGridID].h;
                  }
               }
               else //右边的格子不在计算列表中
               {  MapGrid [MapGrid [_startGridId].rightGridID].
open = true;//加入计算列表中
                  MapGrid [MapGrid [_startGridId].rightGridID].
prevGridID = _startGridId; MapGrid [MapGrid [_startGridId].
rightGridID].g = MapGrid [_startGridId].g + MapGrid[MapGrid
[_startGridId].rightGridID].c;
                  MapGrid[MapGrid[_startGridId].rightGridID].f =
MapGrid [MapGrid [_startGridId].rightGridID].g + MapGrid
[MapGrid[_startGridId].rightGridID].h;
               }
            }
            //在计算列表中寻找估计消耗总值最小的格子
            int NearestId = _startGridId;
            int Least = Int32.MaxValue;
            for (int i = 1; i < MapGrid.Length; i++)
            {  if (MapGrid[i].open == true)
               {  if (MapGrid[i].f < Least)
                  {  NearestId = i;
                     Least = MapGrid[i].f;//估计消耗总值最小
                  }
               }
            }
            //追溯上一步格子
            FindPrev(_endGridId);
            if (NearestId == _startGridId) return;
            AStarFindPath_step(NearestId, _endGridId);
         }
      }
```

### 3.3 Draw 类

```
class Draw
   {  Control Mycontrol;
      Graphics g;
      public static SolidBrush SolidS = new SolidBrush(Color.
Green);//设置开始节点的颜色
      public static SolidBrush SolidP = new SolidBrush(Color.
Blue);//设置路径的颜色
      public static SolidBrush SolidH = new SolidBrush(Color.
Black );//设置障碍节点的颜色
      public static SolidBrush SolidE = new SolidBrush(Color.
Red);//设置结束节点的颜色
```

```
    public static SolidBrush SolidD;//设置背景颜色
    public Draw(Control control)
    {  Mycontrol = control;
            g = control.CreateGraphics();//创建背景控件的
//Graphics 类
       SolidD = new SolidBrush(Mycontrol.BackColor);
    }
    public void DrawStaPoint(int x, int y)
    {  g.FillRectangle(SolidS, x * MapMode.Node + 1, y *
MapMode.Node + 1, MapMode.Node - 1, MapMode.Node
- 1); //绘制起点
    }
    public void DrawEndPoint(int x, int y)
    {  g.FillRectangle(SolidE, x * MapMode.Node + 1, y *
MapMode.Node + 1, MapMode.Node - 1, MapMode.Node
- 1); //绘制终点
    }
    public void DrawHinderPoint(int x, int y)
    {  g.FillRectangle(SolidH, x * MapMode.Node + 1, y *
MapMode.Node + 1, MapMode.Node - 1, MapMode.Node
- 1);  //绘制障碍
    }
    public void DrawPathPoint(int x, int y)
    {  g.FillRectangle(SolidP, x * MapMode.Node + 1, y *
MapMode.Node + 1, MapMode.Node - 1, MapMode.Node
- 1); //绘制路径
    }
    public void ErasePoint(int x, int y)
    {  g.FillRectangle(SolidD, x * MapMode.Node + 1, y *
MapMode.Node + 1, MapMode.Node - 1, MapMode.Node
- 1); //清空节点
    }
    public void BackgroundClear()
    {  //清空游戏背景
       if (Mycontrol ! = null)//如要已载入背景控件
       {  Rectangle rect = new Rectangle(0, 0, Mycontrol.
Width, Mycontrol.Height);
            g.FillRectangle  (new  SolidBrush  (Mycontrol.
BackColor), rect);//用背景色填充背景
       }
    }
    public void DrawBackground()
    {  // 绘制游戏场景
       for  (int i = 0; i <= Mycontrol.Width / MapMode.
Node; i++)//绘制单元格的纵向线
       {  g.DrawLine(new Pen(Color.Black, 1), new Point(i *
MapMode.Node,  0),  new  Point  (i  *  MapMode.Node,
Mycontrol.Height));
       }
       for  (int i = 0; i <= Mycontrol.Height / MapMode.
Node; i++)//绘制单元格的横向线
       {  g.DrawLine(new Pen(Color.Black, 1), new Point(0,
i  *  MapMode.Node),  new  Point  (Mycontrol.Width,  i  *
MapMode.Node));
       }
    }
  }
```

## 4 结语

利用 C# 实现 A* 寻找最短路径算法，通过建立完善的网格数据模型，仅针对网格节点进行计算和维护，使程序流程更加清晰。A* 算法是静态路网中求解最短路径最有效的一种方法，在人工智能寻路、游戏地图寻径等多种场合广泛应用。

（收稿日期：2013-01-10）

（上接第 21 页）

```
  Application.ScreenUpdating = True ' 恢复屏显
End Sub
Private Sub CommandButton3_Click() ' 取消按钮代码
  Unload Me
End Sub
Private Sub TextBox2_Change()
  EvaluateTextEntry TextBox2
End Sub
Private Sub TextBox3_Change()
  EvaluateTextEntry TextBox3
End Sub
Private Sub EvaluateTextEntry(ByVal TheTextBox) ' 输入焦点
' 控制代码
  Application.ScreenUpdating = False ' 关闭屏显
  If Len(TheTextBox) = TheTextBox.MaxLength Then
    If TheTextBox.Name = "TextBox3" Then
      CommandButton1.SetFocus
    Else
      Me.Controls("TextBox" & Val(Mid(TheTextBox.Name, 8,
1)) + 1).SetFocus
    End If
  End If
  Application.ScreenUpdating = True ' 恢复屏显
End Sub
```

## 5 结语

其他窗体和模块的代码文中就不细列了，读者可以从另附的源文件中查阅。本程序在 Windows XP 系统中的 Excel 2003 下测试通过。

（收稿日期：2013-03-29）

# 最优数字分配策略的分析与实现之二

## ——实现违约分最小

尉鹏博　韩银锋

**摘　要**：最优数字分配策略中，当前单元格内违约所扣违约分大，须禁止当前单元格内整数相同和相邻。实现填入值相邻违约最小的同时，须保证该值相邻的相邻违约最小。

**关键词**：当前单元格相等违约；当前单元格相邻违约；违约分

## 1　避免当前单元格内违约

当前单元格内的整数不能相同且不能相邻，如果违反规则，针对每个存储单元都统计一遍，出现一次相同，结果累加100违约分，双向积分共200分；出现一次相邻，结果累加50违约分，双向积分共100分。当前单元格相等违约分200，相当于“相邻单元格相等违约”5个，“相邻单元格相邻违约”10个。当前单元格相邻违约分100，相当于“相邻单元格相等违约”2.5个，“相邻单元格相邻违约”5个。故在此，禁止出现当前单元格内相等、相邻。

（1）判断当前单元格相等违约。本项目中不允许同一格内相等违约存在。判断i行j列单元格第p个元素arr0（1-30的整数）与当前单元格其他元素是否相等违约，返回true不违约，false违约。在类DistributeCount中添加方法：

```
public Boolean validNowEqualData (int i, int j, int p, int arr0)
{Boolean flag = false;//是否违约,默认违约
   int flag1 = 0;//当前单元格及相邻单元格判断违约,默认
//0,1 不违约,2 违约
  int z = 0;//z 从 1-p 个元素
  while (z == 0 || z < p)
  {//==0 是本单元格第 1 个数,flag 为 1
  if (a[i, j, z] ! = arr0) { flag1 = 1; z++; }
    else { flag1 = 2; break; }//违约退出
 }
  if (flag1 <= 1) flag = true;//不违约
  return flag;
}
```

（2）判断当前单元格相邻违约。本项目不允许存在同一格内相邻。判断i行j列单元格第p个元素arr0（1-30的整数）与当前单元格其他元素是否相邻违约，返回true不违约，false违约。

代码与“判断当前单元格相等违约”类似，只需将if (a [i, j, z] ! = arr0) 改为if (a [i, j, z] ! = arr0 - 1 && a [i, j, z] ! = arr0 + 1) 。

## 2　实现相邻违约及相邻的相邻违约最小

相邻违约是填入单元格的值与相邻单元格的值相等或相邻。包括：相邻单元格相等违约，相邻单元格相邻违约。相邻单元格相等违约积20分，双向积40分，根据之前的规定违约类型为2；相邻单元格相邻违约积10分，双向积分20，根据之前的规定违约类型为1。相邻的相邻违约是填入单元格的值与相邻的相邻单元格的值相等，相邻的相邻违约积1分，双向积分2。

要实现相邻违约及相邻的相邻违约最小，是从相邻违约分最低的1-30的几个数字中，选出与之对应的相邻的相邻违约最低的。如有3个数4,10,15的相邻违约最低为0，而4的相邻的相邻违约为6,10的相邻的相邻违约为3,15的相邻的相邻违约为0，则这里选15填入。

（1）相邻违约数组结构

如表1所示。

**表1　相邻违约数组**

| 下标 | 值1-30 | 违约个数 | 违约位置及违约类型（长度为20的数组） | |
|---|---|---|---|---|
| 0 | 1 | 0 | | |
| 1 | 2 | 0 | | |
| 2 | 7 | 0 | | |
| 3 | 18 | 0 | | |
| 4 | 5 | 2 | 1-2-0 | 2 |
| | | | …… | …… |
| | | | | |
| | | | | |
| … | … | | | |
| 29 | 30. | 30 | | |

说明：当前单元格与相邻单元格相邻违约，类型为1，违约个数累计1个；当前单元格与相邻单元格相等违约，类型为2，违约个数累计2个。初始相邻违约数组30个元素，值为

1–30，违约个数为 30。在项目中新建文件夹 entity，entity 中新建类 Valid。代码如下：

```
public class Valid
{ private int validvalue;//值,1–30
    public int Validvalue { get {return validvalue;} set {validvalue=value;}}
    private int validcount;//违约的个数
    public int Validcount {get {return validcount;} set {validcount=value;}}
    private string[,] validarr;
    //违约的位置(下标为 0)及违约类型(下标为 1),0:不违//约,1:相邻违约,2 相等违约
    public string[,] Validarr{get{return validarr;}set{validarr=value;}}
}
```

（2）相邻的相邻违约数组结构

如表 2 所示。

**表 2　相邻的相邻违约数组**

| 下标 | 违约个数 | 违约位置（长度为 40 的数组） |
|---|---|---|
| 0 | 未用 | 未用 |
| 1 | 4 | |
| 2 | 5 | |
| 3 | 1 | 1–3–2 |
| | | |
| | | |
| … | 30 | |
| 7 | 2 | |
| … | 30 | |
| 18 | 0 | |
| … | 30 | |
| 30 | 30 | |

初始相邻的相邻违约数组 31 个元素，0 下标未用，下标对应值 1–30，违约个数为 30。

在 entity 中新建类 ValidNear 类。代码如下：

```
public class ValidNear
{  private int validcount;//违约个数
    public int Validcount { get {return validcount;} set {validcount=value;}}
    private string[] validarr;//违约位置
    public string[] Validarr{ get{return validarr;}set{validarr=value;}}
}
```

（3）初始化相邻违约数组 30 个存储单元，对应 1–30 的值、相应的违约个数，及违约位置：

```
public Valid[] initValid(Valid[] arr)
{ for (int i = 0; i < 30; i++)
  { Valid valid = new Valid();//必须实例化,才可使用
   valid.Validvalue = 0;//违约的值 0,可取 1–30
    valid.Validcount = 30;//违约值对应的违约个数
    valid.Validarr = new string[20, 2];
  //违约值对应的违约位置及违约类型,判断相邻违约的格共
//有 4 个,每个格 5 个元素,因此元素最多 20 个。
    arr[i] = valid;//存入相邻违约数组
}
  return arr;
}
```

（4）初始化相邻的相邻违约数组。共 31 个单元，（0 单元未用，节省计算时间），对应下标即为 1–30 的值。对应存储相邻的相邻违约个数，及违约位置：

```
public ValidNear[] initNearValid(ValidNear[] arr)
{ for (int i = 0; i <= 30; i++)
  { ValidNear validnear = new ValidNear();//必须实例化,
//才可使用
    validnear.Validcount = 30;//违约值(下标)对应的违约
//个数
    validnear.Validarr = new string[40];
    //违约值对应的违约位置,判断相邻违约的格共有 8
//个,因此元素最多 40 个
    arr[i] = validnear;//存入相邻的相邻违约数组
}
  return arr;
}
```

（5）实现相邻违约及相邻的相邻违约最小。

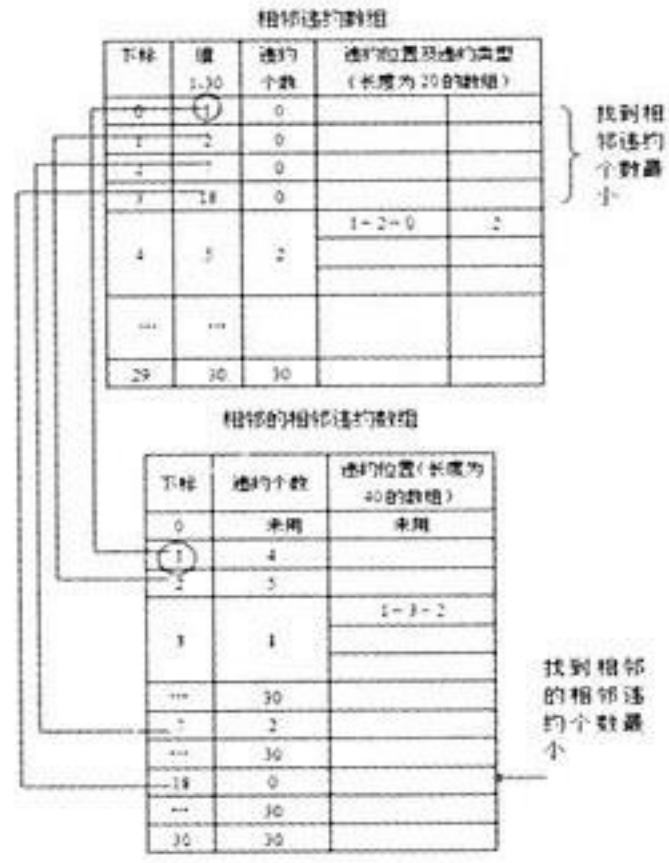



**图 1　实现相邻违约及相邻的相邻违约最小说明**

说明如图 1 所示。初始相邻的相邻违约数组 31 个元素，0 下标未用，下标对应值 1–30，违约个数为 30。相邻违约数组中违约分最小的整数有 1、2、7、18，违约分为 0。1 对应相邻的相邻违约分，是在相邻的相邻违约数组中找下标为 1 的，其违约个数为 4。2 对应相邻的相邻违约分，是在相邻的相邻

违约数组中找下标为 2 的，其违约个数为 5。7 对应相邻的相邻违约分，是在相邻的相邻违约数组中找下标为 7 的，其违约个数为 2。18 对应相邻的相邻违约分，是在相邻的相邻违约数组中找下标为 18 的，其违约个数为 0。这里 18 的相邻违约最小，且 18 的相邻的相邻违约也最小，所以选择填入整数为 18。

1）比较当前值 arr0（1–30 的整数）与（x,y）单元格内的数据是否相邻违约。validabsolut 相邻违约数组，count 违约个数，m 违约下标：

```
private void validDataUnit(int x, int y, int arr0, ref string[,]
validabsolut, ref int count, ref int m)
{ for (int z = 0; z < 5; z++)
  { if (a[x, y, z] == -1) { break; }//判断 2–5 个元素,未填
//入的为–1,不用比
   else
  { if (a[x, y, z] == arr0)//判断相邻格是否相等违约
 { validabsolut[m, 0] = x + "–" + y + "–" + z;//记录位置
      validabsolut[m, 1] = "2";//记录类型
          count = count + 2;//相等违约相当于相邻违
//约 2 个
      m++;//违约位置下标加 1
}
      if(a[x, y, z]==arr0–1||a[x, y, z]==arr0+1)//判断相邻
//格是否相邻违约
      { validabsolut[m, 0] = x + "–" + y + "–" + z;//记录
//位置
        validabsolut[m, 1] = "1";//记录类型
        count++;//相邻违约累计 1 个
 m++;//违约位置下标加 1
      }
 }
}
}
```

2）将违约信息插入相邻违约顺序表，按违约个数排序：

```
public Valid[] insertArr(Valid[] arr, int x, int y, int k, string
[,] alidpost)
{  if (k == 0)//第一个数直接插入
   {  arr[0].Validvalue = x;//违约值
      arr[0].Validcount = y;//违约个数
        arr[0].Validarr = validpost;//长度为 20 的违约位置
//数组
  }
   else//插入顺序表,查找插入位置,第一个大于的就停止,
//按违约个数排序
   { int i = 0;
     for (; i < 30; i++)
      { if (arr[i].Validcount > y) { break; }//查找应该插入
//的位置 }
 int m = i;//防止是下标为 0 的元素
     for (int j = k – 1; j >= m; j--) //从 i–k 均后移
 { arr[j + 1].Validvalue = arr[j].Validvalue;
   arr[j + 1].Validcount = arr[j].Validcount;
   arr[j + 1].Validarr = arr[j].Validarr;
 }
 //插入
 arr[m].Validvalue = x;
 arr[m].Validcount = y;
 arr[m].Validarr = validpost;
   }
  return arr;
 }
```

（6）查找相邻违约个数最小及相邻的相邻违约个数最小：

```
private int findMinNear()
{int minvalid = validarr[0].Validcount; //取相邻违约数组中
//违约个数最小的
 //取与相邻最小对应的相邻的相邻中最小的下标,即 1–30
//的值
 int minnear = validarr[0].Validvalue;
 //取相邻违约个数最小的值的相应相邻的相邻违约个数
 int minnearvalid = validneararr[minnear].Validcount;
 varvalid = validarr[0]; //初始化最小相邻违约
 varvalidnear = validneararr[validarr[0].Validvalue];
 //初始化最小相邻的相邻违约
 for (int m = 1; m < 30; m++)
  {if (validarr[m].Validcount <= minvalid)
   //找到满足最小相邻违约的 1–30 的值
   {if (minnearvalid > validneararr [validarr [m].Validvalue].
Validcount)
      {//找到满足最小相邻的相邻违约的 1–30 的值
       varvalid = validarr[m];
        minnear = validarr[m].Validvalue;
          varvalidnear = validneararr [validarr [m].
Validvalue];
          minnearvalid = validneararr [validarr [m].
Validvalue].Validcount;
       }
     }else{break; }
 }
   return minnear;
 }
```

## 参考文献

[1] Nick Randolph. Visual Studio2010 高级编程 [M]. 清华大学出版社，2012，01.

[2] 邱仲潘. Visual Basic 2010 (中文版) 从入门到精通 [M]. 电子工业出版社，2011，01.

[3] 汪沁. 数据结构 [M]. 清华大学出版社，2009，09.

（收稿日期：2013–04–02）

# 基于 Delphi 的报表统计系统设计与实现

武 伟

**摘 要**：利用 Delphi、Excel 等工具相结合办法来实现钢铁公司生产日报的报表系统设计。

**关键词**：Delphi 语言；报表；Excel 工具

## 1 引言

在报表系统实际实现时，有许多报表生成工具，如 Microsoft 的 CrystalReport，报表生成工具的功能越来越强大，但是由于在现实应用中的千差万别，客户需求不尽相同，使得各种报表工具在开发应用中侧重点各不相同。文中使用 Delphi、Excel 等工具相结合的方式来实现报表系统的设计，而这种方式在实践中也得到了很好的应用。

## 2 报表系统设计与实现

本报表系统基于 C/S 模式，报表的数据从数据库中提取。系统包括以下几个模块：数据库的设计，数据的采集与处理，报表内容的填充与汇总。下面以钢铁公司日常生产报表为例，来看一下报表的具体实现。

### 2.1 数据库

数据库采用 SQL Server 2000，钢铁公司生产报表主要包括铁水、钢水、各种的钢材日生产量、日计划量、月累计产量等主要信息，数据库表结构如图 1 所示。

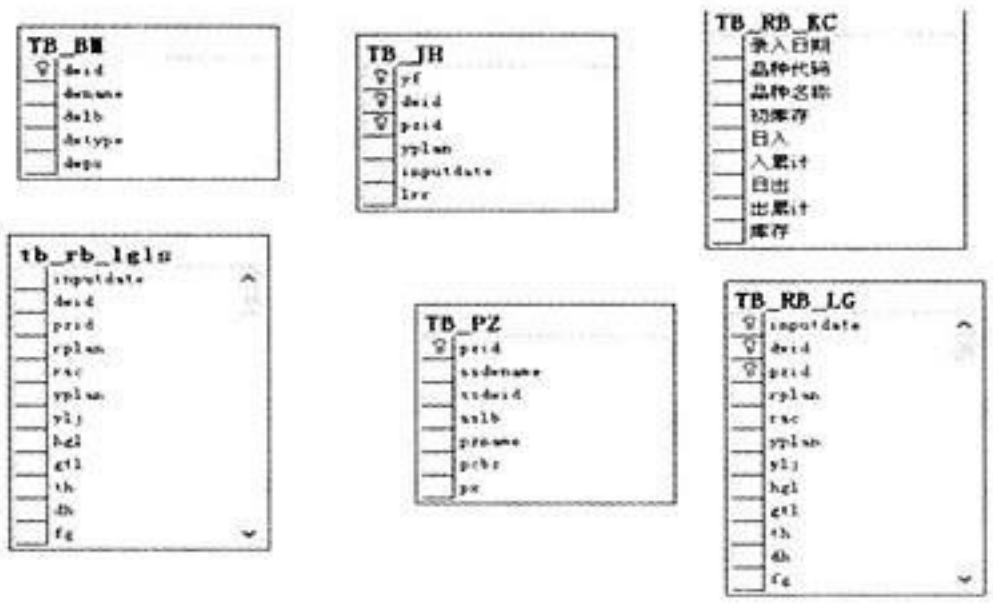



图 1 数据的表结构

### 2.2 数据采集与处理

数据采集与处理主要是把报表所需要数据经过计算、汇总等方式予以处理，然后将数据保存到数据库中。本系统采用 Delphi 开发工具来实现对数据的采集与处理。在“ODBC 数据源管理器”中添加数据源，与数据库相连。下面是手工录入功能界面，如图 2 所示。

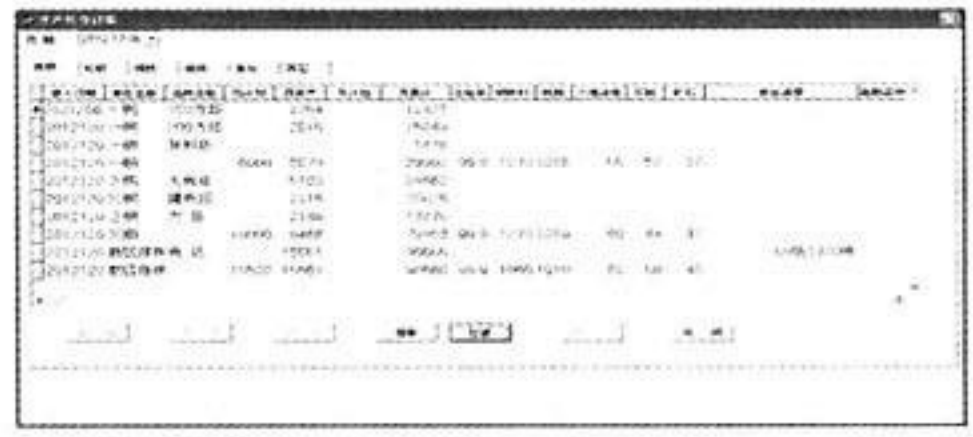


图 2 生成数据录入界面

这是炼钢生产数据的录入界面，主要由 PageContro 控件组成，下列控件与数据源实现绑定：数据源—Ttable 控件—DataSource 控件—dbgrid 控件；数据源—Query 控件。

这样 dbgrid 控件就实现对数据的录入与修改。

（1）点击“提取”按钮，将前一天中像“品种名称”、“日计划”等固定数据提出，这样可减少数据输入的工作量，实现代码如下：

```
procedure TForm_RBb.Button_sjreadClick(Sender: TObject);
var
i,j,k :integer;
begin
j:= strtoint(formatdatetime('dd',dtp1.datetime));
    with query0 do
     begin
       close;
       sql.clear;
   sql.add('select * from tb_rb_sj where inputdate=:pdate');
         parambyname  ('pdate').asstring:  =formatdatetime
('yyyymmdd',dtp1.datetime-1);
       open;
     end;
     i:=0;
       for k:=1 to j do
       begin
        i:=i+1;
 with query0 do
  begin
    close;
```

```
        sql.clear;
 sql.add('select * from tb_rb_sj where inputdate=:pdate');
          parambyname   ('pdate').asstring:  =formatdatetime
('yyyymmdd',dtp1.datetime-i);
        open;
      end;
    if query0.recordcount>0 then
      begin
        break;
      end;
      end;
     with query0 do
      begin
        close;
        sql.clear;
 sql.add('select * from tb_rb_sj where inputdate=:pdate');
          parambyname   ('pdate').asstring:  =formatdatetime
('yyyymmdd',dtp1.datetime);
        open;
      end;
       if query0.recordcount>0 then
      begin
       application.MessageBox('当天已有数据,不可覆盖','提
示信息',MB_OKCANCEL);
       Button_sjfilter.Click ;
        exit;
      end;
    with query0 do
      begin
        close;
        sql.clear;
        sql.add('delete from tb_rb_sjls');
        execsql;
      end;
     with query0 do
      begin
        close;
        sql.clear;
     sql.add('insert into tb_rb_sjls(inputdate,pzid,rplan) select
inputdate,pzid,rplan from tb_rb_sj where inputdate=:p');
                parambyname   ('p').asstring:  =formatdatetime
('yyyymmdd',dtp1.datetime-i);
        execsql;
      end;
     with query0 do
      begin
        close;
        sql.clear;
        sql.add('update tb_rb_sjls set inputdate=:p');
                 parambyname   ('p').asstring: =formatdatetime
('yyyymmdd',dtp1.datetime);
        execsql;
      end;
    with query0 do
      begin
        close;
        sql.clear;
        sql.add('insert into tb_rb_sj select * from tb_rb_sjls');
       execsql;
      end;
    with query0 do
      begin
        close;
        sql.clear;
        sql.add('delete from tb_rb_sjls');
       execsql;
      end;
     tb_sjls.close;
     tb_sjls.open;
    button_sjsave.enabled:=true;
    tb_sj.active:=false;
   tb_sj.Filtered:=false;
    tb_sj.Filter: ='inputdate =' +formatdatetime ('yyyymmdd',
dtp1.datetime)+'';
    tb_sj.Filtered:=true;
    tb_sj.active:=true;
    button_sjadd.enabled:=true;
     button_sjdel.enabled:=true;
   button_sjsave.Enabled:=true;
end;
```

（2）点击“保存”，自动完成日产量累计、月产量累计的计算并保存到数据库中，实现代码如下：

```
procedure TForm_RBb.Button_lgsaveClick(Sender: TObject);
var
 str1,str2,str3:string;
begin
  try
    begin
    tb_lg.EditKey;
    tb_lg.ApplyUpdates;
    end;
 except
  begin
showmessage('数据有误,请检查数据的合法性和唯一性!');
    exit;
  end;
end;
   with query0 do
          begin
           close;
           sql.clear;
           sql.add ('delete from tb_rb_lg where pzid=222 and
inputdate=:pdate' );
```

```
            parambyname   ('pdate').asstring: =formatdatetime
('yyyymmdd',dtp1.DateTime);
            execsql;
            end;
  with query_lg do
    begin
     close;
     sql.clear;
      sql.add ('select * from tb_rb_lg where inputdate=:pdate
and pzid<>:pzid');
       parambyname   ('pdate').asstring: =  formatdatetime
('yyyymmdd',dtp1.DateTime);
      parambyname('pzid').asstring:='111111';
      open;
     end;          //选择当天所有产品的纪录
   for irow:=1 to query_lg.RecordCount do
     begin
        with query0 do
         begin
          close;
          sql.clear;
       sql.add('select sum(rsc) ylj from tb_rb_lg where pzid=:
pzid and inputdate>=:pdate1 and inputdate<=:pdate2' );
           parambyname  ('pdate1').asstring: =formatdatetime
('yyyymm',dtp1.DateTime)+'01';
                    parambyname  ('pzid').asstring:=query_lg.
FieldByName('pzid').asstring;
           parambyname  ('pdate2').asstring: =formatdatetime
('yyyymmdd',dtp1.DateTime);
          open;
         end;
        with query1 do
         begin
          close;
          sql.clear;
          sql.add ('update tb_rb_lg set ylj=:pylj where pzid=:
pzid and inputdate=:pdate');
       parambyname  ('pzid').asstring: =query_lg.FieldByName
('pzid').asstring;
       parambyname('pdate').asstring:=query_lg.FieldByName
('inputdate').asstring;
     parambyname  ('pylj').asinteger:=query0.fieldbyname('ylj').
asinteger;
           execsql;
          end;
        query_lg.Next;
      end;    //逐一更新当天每个产品的月累计
  with query_lg do
    begin
      close;
      sql.clear;
      sql.add('select sum(rsc) rsc,dwid  from tb_rb_lg where
pzid<>:pzid and inputdate=:pdate group by dwid');
       parambyname('pzid').asstring:='111111';
         parambyname   ('pdate').asstring:   =formatdatetime
('yyyymmdd',dtp1.DateTime);
       open;      //计算各单位的日实产合计
     end;
   while not  query_lg.eof do
    begin
      with query0 do
        begin
          close;
          sql.clear;
           sql.add ('update tb_rb_lg set rsc = :prsc where
pzid=:pzid and dwid=:pdwid and inputdate=:pdate');
            parambyname   ('prsc').asinteger:   =query_lg.
fieldbyname('rsc').asinteger;
            parambyname   ('pdwid').asinteger:  =query_lg.
fieldbyname('dwid').asinteger;
           parambyname  ('pdate').asstring:=formatdatetime
('yyyymmdd',dtp1.DateTime);
           parambyname('pzid').asstring:='111111';
           execsql;
         end;
      query_lg.next;
    end;      //更新每个单位日实产合计
   with query_dwlj do
    begin
      close;
      sql.clear;
       sql.add('select sum(rsc) ylj,dwid  from tb_rb_lg where
pzid=:pzid and inputdate>=:pdate1 and inputdate<=:pdate2
group by dwid');
       str1:=formatdatetime('yyyymm',dtp1.DateTime)+'01';
      parambyname('pdate1').asstring:=str1;
      parambyname('pzid').asstring:='111111';
      str2:=formatdatetime('yyyymmdd',dtp1.DateTime);
      parambyname('pdate2').asstring:=str2;
      open;
    end; //计算每个单位的月累计
  while not  query_dwlj.eof do
    begin
      with query0 do
        begin
          close;
          sql.clear;
          sql.add('update tb_rb_lg set ylj=:pylj where pzid=:
pzid and dwid=:pdw and inputdate=:pdate');
            parambyname   ('pylj').asinteger:  =query_dwlj.
fieldbyname('ylj').asinteger;
            parambyname   ('pdw').asinteger:  =query_dwlj.
fieldbyname('dwid').asinteger;
           parambyname  ('pdate').asstring:=formatdatetime
```

```
('yyyymmdd',dtp1.DateTime);
            parambyname('pzid').asstring:='111111';
            execsql;
          end;
       query_dwlj.next;
    end;
  tb_lg.Close;
  tb_lg.open;
  button_lgadd.enabled:=true;
   cb_dwlg.Visible:=false;
   cb_pzlg.Visible:=false;   //更新各单位的月累计
  showmessage('存盘成功!');
  button_adjust.enabled:=true;
end;
```

## 2.3 报表内容的填充与汇总

利用 F1book 控件完成报表基本表格格式设计，然后将数据库中的数据填充到 F1book 中的表格，最好将 F1book 中的报表按照指定格式导出生成 Excel 文件。

（1）往 F1book 中填充数据，下面为部分代码：

```
//////////////////////
  f1book1.setfont ('宋体',b,true,false,false,false,rgb(0,0,0),
false,false);
  f1book1.TextRC[2,1]:='炼 钢';
  f1book1.TextRC[2,2]:='日目标';
  f1book1.TextRC[2,3]:='日实产';
  f1book1.TextRC[2,4]:='月目标';
  f1book1.TextRC[2,5]:='月累计';
  f1book1.TextRC[2,6]:='合格率';
  f1book1.TextRC[2,7]:='钢铁料';
  f1book1.TextRC[2,8]:='铁耗';
  f1book1.TextRC[2,9]:='少渣冶炼';
  f1book1.TextRC[2,10]:='灰耗';
  f1book1.TextRC[2,11]:='矿石';
  ////////////////////////////////
    with query_lg_zb do
    begin
      close;
      sql.Clear;
       sql.add ('select * from tb_rb_lg where inputdate=:
pdate and dwid=:pdwid  and pzid=:pzid');
        ParamByName ('pdate').asstring: =formatdatetime
('yyyymmdd',DTP1.DateTime);
       ParamByName('pzid').asstring:='111111';
      ParamByName('pdwid').asstring:='0101';
      open;
    end;
 with query0 do
    begin
      close;
      sql.Clear;
          sql.add('select * from tb_jh where yf =:pdate and
dwid=:pdwid and pzid=:pzid');
        ParamByName ('pdate').asstring: =formatdatetime
('yyyymm',DTP1.DateTime);
      ParamByName('pzid').asstring:='111111';
      ParamByName('pdwid').asstring:='0101';
      open;
    end;
     i:=3;
      f1book1.Selection:='a'+inttostr(i);
f1book1.setalignment(f1halignleft, false, f1valigncenter, 0);
       f1book1.TextRC[i,1]:='一炼钢';
  if query_lg_zb.fieldbyname('rplan').asstring <> '0' then
       begin
         f1book1.TextRC [i,2]: =query_lg_zb.fieldbyname
('rplan').asstring;  //yplan
       end;
 lgrjh:= lgrjh + query_lg_zb.fieldbyname('rplan').asinteger;
f1book1.TextRC[i,3]:=query_lg_zb.fieldbyname('rsc').asstring;
lgrsc:= lgrsc + query_lg_zb.fieldbyname('rsc').asinteger;
f1book1.TextRC[i,4]:=query0.fieldbyname('yplan').asstring;
      lgyjh:= lgyjh+ query0.fieldbyname('yplan').asinteger;
f1book1.TextRC[i,5]:=query_lg_zb.fieldbyname('ylj').asstring;
      lgylj:= lgylj + query_lg_zb.fieldbyname('ylj').asinteger;
     rplana:=0;
   with query_rjh do
     begin
      close;
      sql.clear;
      sql.add('select * from tb_rb_lg where pzid=:pzid  and
dwid= :dwid and inputdate<= :date  and inputdate>=:date1
order by inputdate');//一钢数据加载
      parambyname('pzid').asstring:='111111';
      parambyname('dwid').asstring:='0101';
       parambyname ('date').asstring: =formatdatetime
('yyyymmdd',DTP1.DateTime);
       parambyname ('date1').asstring: =formatdatetime
('yyyymm',DTP1.DateTime)+'01';
      open;
     end;
     while not query_rjh.eof  do
       begin
  rplana:= rplana + query_rjh.fieldbyname('rplan').AsInteger;
        query_rjh.next;
       end;
f1book1.TextRC[i,6]:=query_lg_zb.fieldbyname('hgl').asstring;
f1book1.TextRC[i,7]:=query_lg_zb.fieldbyname('gtl').asstring;
f1book1.TextRC[i,8]:=query_lg_zb.fieldbyname('th').asstring;
f1book1.TextRC[i,9]:=query_lg_zb.fieldbyname('fg').asstring;
f1book1.TextRC[i,10]:=query_lg_zb.fieldbyname('hh').asstring;
f1book1.TextRC[i,11]:=query_lg_zb.fieldbyname('ks').asstring;
      i:=i+1;
   with query0 do
```

```
    begin
      close;
      sql.Clear;
      sql.add ('select * from tb_pz where ssdwid=:pdwid
and pzid<>:pzid order by pzid');
      ParamByName('pdwid').asstring:='0101';
      ParamByName('pzid').asstring:='111111';
      open;
     end;
    while not query0.Eof  do
      begin
       with query_lg_cb do
        begin
         close;
         sql.Clear;
         sql.add ('select * from tb_rb_lg where inputdate=:
pdate and dwid=:pdwid and pzid=:ppzid');
          ParamByName ('pdate').asstring:=formatdatetime
('yyyymmdd',DTP1.DateTime);
          ParamByName('pdwid').asstring:='0101';
                ParamByName ('ppzid').asstring:=query0.
fieldbyname('pzid').asstring;
          open;
        end;
       with query00 do
        begin
         close;
         sql.Clear;
         sql.add ('select * from tb_jh where yf=:pdate and
dwid=:pdwid and pzid=:ppzid');
          ParamByName ('pdate').asstring:=formatdatetime
('yyyymm',DTP1.DateTime);
          ParamByName('pdwid').asstring:='0101';
       ParamByName ('ppzid').asstring:=query0.fieldbyname
('pzid').asstring;
          open;
        end;
        with query1 do
         begin
           close;
           sql.Clear;
           sql.add ('select max (ylj) ylj from tb_rb_lg where
inputdate<=:pdate1  and inputdate>=:pdate2 and dwid=:
pdwid and pzid=:ppzid');
          ParamByName ('pdate1').asstring:=formatdatetime
('yyyymmdd',DTP1.DateTime);
          ParamByName ('pdate2').asstring:=formatdatetime
('yyyymm',DTP1.DateTime)+'01';
            ParamByName('pdwid').asstring:='0101';
 ParamByName('ppzid').asstring:=query0.fieldbyname('pzid').
asstring;
            open;
          end;
          f1book1.Selection:='a'+inttostr(i);
 f1book1.setalignment(f1halignright,false, f1valigncenter, 0);
  f1book1.TextRC  [i,1]: =" +getcpname (QUERY_cpNAME,
query0.fieldbyname('pzid').asstring);
     // f1book1.TextRC[i,1]:=getcpname(QUERY_cpNAME,
query0.fieldbyname('pzid').asstring);
f1book1.textRC[i,3]:=query_lg_cb.fieldbyname('rsc').asstring;
 f1book1.textRC[i,4]:=query00.fieldbyname('yplan').asstring;
 f1book1.textRC[i,5]:=query1.fieldbyname('ylj').asstring;
      f1book1.Selection:='a'+inttostr(i);
        query0.Next;
        i:=i+1;
      end;
```

（2）将数据按照日期导出生成 Excel 文件，并对其进行格式化处理，部分代码如下：

```
 ExcelApp:=CreateOleObject('Excel.Application');
  ExcelApp.Caption:=' 应用程序调用 Microsoft Excel';
   ExcelApp.workBooks.Open  ('c:\zd\ 生 产 日 报 ' +
formatdatetime('yyyy-mm-dd',dtp1.DateTime)+'.xls');
//打开已存在工作簿
 ExcelApp.Worksheets[1].activate;
 ExcelApp.DisplayAlerts  :=  False;
 m:=44;
 excelapp.range ['B'+inttostr (m)].value:=trim(excelapp.range
['B'+inttostr (m)].value +excelapp.range ['B'+inttostr (m+1)].
value) ;   //不出错误提示框
 excelapp.range['B'+inttostr(m)+':B'+inttostr(m+1)].Merge;
  excelapp.range  ['B' +inttostr  (m) +':B' +inttostr  (m +1)].
horizontalalignment:=xlcenter;
  excelapp.range  ['B' +inttostr  (m) +':B' +inttostr  (m +1)].
verticalalignment:=xlcenter;
 excelapp.range ['D'+inttostr (m)].value:=trim(excelapp.range
['D'+inttostr (m)].value +excelapp.range ['D'+inttostr (m+1)].
value) ;
 excelapp.range['D'+inttostr(m)+':D'+inttostr(m+1)].Merge;
  excelapp.range  ['D' +inttostr  (m) +':D' +inttostr  (m +1)].
horizontalalignment:=xlcenter;
  excelapp.range  ['D' +inttostr  (m) +':D' +inttostr  (m +1)].
verticalalignment:=xlcenter;
 ExcelApp.Worksheets[2].activate;
 ExcelApp.DisplayAlerts  :=  False;
```

## 3 结语

目前，该报表系统在实际应用中效果不错，虽然用老的开发工具，代码繁多，但是能够满足个性用户的需求。当然，这还需要进一步完善。

（收稿日期：2013-03-25）

# 高三年级调研测试数据采集与分析

马浩洲　陈会芹

**摘　要**：为帮助各校了解高三教学效果，提供更多的样本参照和有效的教学策略实施依据，多年来对高三年级调研测试。在调研中运用 VFP 进行数据采集、分析与处理，大大提高了样本的处理效率，为提高教学效果提供了重要依据。

**关键词**：调研测试；数据；采集；分析

## 1　引言

江苏省普通高考模式为“3+学业水平测试+综合素质评价”。统考科目为语文、数学、外语 3 门。各科分值设定为：语文 160 分，数学 160 分，外语 120 分，共 440 分。语文、数学分别另设附加题 40 分。文科类考生加试语文附加题；理科类考生加试数学附加题；不兼报文科类或理科类专业的体育类、艺术类考生不加试附加题。文科类、理科类考生 3 门统考总分为 480 分，体育类、艺术类考生 3 门统考总分为 440 分。必测科目（物理、化学、生物、政治、历史及地理 6 门）各科满分为 100 分，按考生得分分为 A、B、C、D 4 个等级。本科高校的等级一般要达到 2B。由于江苏高考的特殊性，在调研中数据的采集、分析与处理也有其特殊性，一般的通用软件满足不了调研的需求，需要编写专用程序，提高工作效率。

## 2　实现

根据调研考试要求及单位现有软硬件条件，本系统用 VFP 编写程序，实现考前数据采集和考后数据分析与处理功能。考前数据采集主要功能有：表的建立、各学校数据合并、对表中数据的有效性进行检查。考后数据分析与处理主要功能有：核查 6 门成绩、学校数据拆分、语数外 3 门总分、文理分科、语数外 3 门平均分、选修科目各等级达标人数、选修科目平均分、总分分数段统计、选修六科达 B 人数、主副卷平均分统计、县模拟录取统计、区县上线人数及四星级学校上线人数等。

## 3　数据表结构

系统设计所需数据表 Lsx.dbf 和 dj.dbf 结构分别如表 1 和表 2 所示。

表 1　Lsx.dbf

| 字段名称 | 字段类型 | 字段长度 | 中文含义 |
|---|---|---|---|
| xqmc | 字符型 | 6 | 县区名称 |
| xqdm | 字符型 | 2 | 县区代码 |
| xxmc | 字符型 | 20 | 学校名称 |
| xxdm | 字符型 | 4 | 学校代码 |
| xxxj | 字符型 | 1 | 学校星级 |
| kch | 字符型 | 2 | 考场号 |
| zwh | 字符型 | 2 | 座位号 |
| zkzh | 字符型 | 9 | 准考证号 |
| bjh | 字符型 | 2 | 班级号 |
| kslb | 字符型 | 1 | 考生类别 |
| xkmc | 字符型 | 4 | 选科名称 |
| xkdm | 字符型 | 1 | 选科代码 |
| xm | 字符型 | 8 | 姓　名 |

表 2　dj.dbf

| 字段名称 | 字段类型 | 字段长度 | 中文含义 |
|---|---|---|---|
| xk | 字符型 | 4 | 学科名称 |
| cks | 数值型 | 5 | 学科参考人数 |
| aa | 数值型 | 3 | A+等级分数线 |
| a | 数值型 | 3 | A 等级分数线 |
| bb | 数值型 | 3 | B+等级分数线 |
| b | 数值型 | 3 | B 等级分数线 |
| c | 数值型 | 3 | C 等级分数线 |
| d | 数值型 | 3 | D 等级分数线 |

## 4　主要代码

### 4.1　考前数据采集

（1）建立表“学校简称.db”f 表 (我县 5 所高中简称分别为 lz、zg、yz、lx、jc)：

```
create table 学校简称(xqmc  c(6), xqdm  c(6), ;
xxmc  c(20), xxdm  c(4) , xxxj c(1), kch c(2), ;
zwh  c(2), zkzh c(9) , bjh c(2), kslb c(1), ;
xkmc  c(4), xkdm c(1) , xm c(8))
```

（2）各校按照上表的要求录入学生信息，生成 dbf 表。为了满足阅卷的需求，需要对各校的表进行合并，生成 lsx.dbf：

```
use lsx
append  from  lz
append  from  zg
append  from  yz
append  from  lx
append  from  yz
append  from  jc
```

为了考后录分及数据处理的准确、高效，还需要对合并后表中数据的有效性进行检查。

(3) 检查字段内容是否为空：

```
select * from lsx where empty (xqmc) or empty(xqdm) or empty(xxmc) or empty(xxdm) or empty(xxxj) or empty (kch) or empty (zwh) or empty (zkzh) or empty (bjh) or empty(kslb) or empty(xkmc) or empty(xkdm) or empty(xm)
```

(4) 为了分发试卷的方便，需检查 xkmc (选课名称) 和 xkdm (选课代码) (物化 1，物生 2，物政 3，物地 4，历化 5，历生 6，历政 7，历地 8，艺体 9) 是否对应。添加 xkdm1 字段，利用表中原有的 xkmc 字段数据添加与之相对应的数据到 xkdm1 字段中，比较 xkdm1 与 xkdm 字段内容是否相等：

```
SET TALK OFF
SET SAFE OFF
CLOS DATA
use lsx
*alter table lsx add xkdm1 c(1)
repl xkdm1 with "0" all
repl xkdm1 with "1" for xkmc="物化" all
repl xkdm1 with "2" for xkmc="物生" all
repl xkdm1 with "3" for xkmc="物政" all
repl xkdm1 with "4" for xkmc="物地" all
repl xkdm1 with "5" for xkmc="历化" all
repl xkdm1 with "6" for xkmc="历生" all
repl xkdm1 with "7" for xkmc="历政" all
repl xkdm1 with "8" for xkmc="历地" all
repl xkdm1 with "9" for xkmc="艺体" all
select *  from lsx where  xkdm! = xkdm1  or  xkdm1="0"
CLOS DATA
RETU
```

(5) 检查县区名称、县区代码、学校名称、代码、星级、考生类别、考场号及座位号：

```
select *  from lsx where  xqmc! = "某县"  or  xqdm ! = "06"
SELECT xxmc, xxdm,xxxj,kslb  DISTINCT  from  lsx
SELECT kch,zwh  DISTINCT  from  lsx
```

(6) 检查准考证号。准考证号第 1 位为年份，用 3 表示；第 2、3 位为县区代码；第 4、5 位为学校代码，第 6-9 位为考生流水号 (从 0001 号起，连续编排)：

```
Select * from lsx where subs (zkzh,1,1) ! ="3" or subs (zkzh,2,2) ! ="06" or subs(zkzh,2,4) ! = xxdm
```

(7) 检查准考证是否重号：

```
select  zkzh as  准考证号 ,count(*)  as  准考证重复次数;
from lsx;
group by zkzh;
having count(*)>1
```

## 4.2 考后数据分析功能的实现

(1) 把全县总表中的数据进行拆分成以学校名称命名的多张表，下发到各学校，便于各学校统计本校成绩：

```
clos data
sele 1
use lsx EXCLUSIVE
index on xxmc tag xxmc uniq
copy to 学校名称表 field xxmc
set order to
sele 2
use 学校名称表
do whil ! eof()
  mc=xxmc
  sele 1
  copy to &mc for xxmc=mc
  sele 2
  skip
enddo
```

(2) 为了成绩的可比性，成绩分析与处理分为县理科、县文科、县理科应届生、县理科往届生、县文科应届生、县文科往届生等类别。下文只列举理科应届 (下文简称“理应”) 成绩处理代码，其他代码类似：

```
Select xxmc as 学校名称,count(*) as 学校理应参考人数,;
avg(yw +  ywfj) as 语文平均分 , avg( sx+ sxfj) as 数学平均分 ,avg( yy) as 英语平均分 ,avg( yw+ywfj+sx+sxfj+yy) as 总分平均分 ;
from  lsx.dbf;
where xkmc like '%物%' and kslb= '1' and zf1>0 ;
group by xxmc;
order by xxdm ASC;
INTO TABLE 学校理应三科平均分.dbf
copy to  学校理应三科平均分.xls  type  xl5
Select count(*) as  县理应参考人数,;
avg(yw+ywfj ) as 语文均分, avg( sx+ sxfj) as 数学均分, avg( yy) as 英语均分 ,avg( yw+ywfj+sx+sxfj+yy) as 总分总均分 ;
from  lsx.dbf;
where xkmc like '%物%' and kslb= '1'and zf1>0 ;
INTO TABLE  县理应三科平均分.dbf
copy to  县理应三科平均分.xls  type  xl5
```

(3) 物理选修科目各等级人数，其他科目代码类似：

```
SET TALK OFF
SET SAFE OFF
CLOS DATA
sele 1
```

```
use dj
sele 2
use lsx
sele 1
go 1
a1=aa
a2=a
b1=bb
b2=b
c1=c
d1=d
sele 2
Select xxmc as 学校名称  ,count (*) as  学校参考总人数,;
sum(iif(wl>0,1,0)) as 物理参考人数,;
sum(iif(wl>=a1,1,0)) as A 加级,;
sum(iif(wl>=a2 and wl<a1  ,1,0)) as A 级,;
sum(iif(wl>=b1 and wl<a2  ,1,0)) as B 加级,;
sum(iif(wl>=b2 and wl<b1  ,1,0)) as B 级,;
sum(iif(wl>=c1 and wl<b2  ,1,0)) as C 级,;
sum(iif(wl>=d1 and wl<c1  ,1,0)) as D 级,;
sum(iif(wl>=d1 ,1,0)) as  合计;
from  lsx;
group by xxmc;
order by xxdm ASC;
INTO TABLE  物理选修科目各等级人数.dbf
copy to  物理选修科目各等级人数.xls  type  xl5
```

（4）物理选修科目平均分：

```
Select xxmc as 学校名称  ,count(*) as  学校参考人数,;
sum(iif(wl>0,1,0)) as 物理参考人数,;
avg(wl) as  物理;
from  lsx;
where wl>0;
group by xxmc;
order by xxdm ASC;
INTO TABLE  学校物理平均分.dbf
copy to  学校物理平均分.xls  type  xl5
Select  sum(iif(wl>0,1,0)) as 县物理参考人数,;
avg(wl) as  物理;
from  lsx;
where wl>0;
INTO TABLE  县物理平均分.dbf
copy to  县物理平均分.xls  type  xl5
```

（5）总分分数段统计：

```
Select xxmc as 学校名称 ,count  (*) as  学校理应参考人数,;
sum(iif((yw+ywfj+sx+sxfj+yy)>=a1,1,0)) as a,;
sum(iif((yw+ywfj+sx+sxfj+yy)>=a2,1,0)) as b,;
sum(iif((yw+ywfj+sx+sxfj+yy)>=a3,1,0)) as c,;
sum(iif((yw+ywfj+sx+sxfj+yy)>=a4,1,0)) as d,;
sum(iif((yw+ywfj+sx+sxfj+yy)>=a5,1,0)) as e,;
sum(iif((yw+ywfj+sx+sxfj+yy)>=a6,1,0)) as f,;
sum(iif((yw+ywfj+sx+sxfj+yy)>=a7,1,0)) as g,;
sum(iif((yw+ywfj+sx+sxfj+yy)>=a8,1,0)) as h,;
sum(iif((yw+ywfj+sx+sxfj+yy)>=a9,1,0)) as i;
from  lsx.dbf;
where xkmc like '%物%' and kslb= '1' ;
group by xxmc;
order by xxdm ASC;
INTO TABLE  学校理应分数段统计.dbf
copy to  学校理应分数段统计.xls  type  xl5
```

（6）物理选修科目达 B 人数：

```
Select xxmc as 学校名称  ,count(*) as  学校参考人数,;
sum(iif(wl>0,1,0)) as 物理参考人数,;
sum(iif(wl>=wlb  ,1,0)) as B 级;
from lsx;
group by xxmc;
order by xxdm ASC;
INTO TABLE 物理选修科达 B 人数.dbf
copy to  物理选修科达 B 人数.xls  type  xl5
```

（7）学校语文平均分，其他科目代码类似：

```
Select xxmc as  学校名称 ,count(*) as  学校参考人数,;
avg(yw ) as 语文主卷平均分;
from  lsx.dbf;
where  yw>0 ;
group by xxmc;
order by xxdm ASC;
INTO TABLE A1 学校语文主卷平均分.dbf
copy to A1 学校语文主卷平均分分.xls  type  xl5
Select xxmc as  学校名称 ,count(*) as  学校参考人数,;
avg(ywfj ) as 语文副卷分;
from  lsx.dbf;
where  ywfj>0 ;
group by xxmc;
order by xxdm ASC;
INTO TABLE  学校语文副卷评分分.dbf
copy to  学校语文副卷平均分.xls  type  xl5
```

（8）市教研室公布模拟录取本科分数线（理科为 320 分，选修科目达 2B），统计县理应模拟录取各校上线人数：

```
Select  xxmc  as 学校名称 ,count(*) as  理应参考人数,;
sum(iif((yw+ywfj+sx+sxfj+yy)>=320,1,0)) as 理应录取人数,;
sum(iif(b2=2,1,0)) as 理应达 2B;
from  lsx.dbf;
where xkmc like '%物%' and kslb= '1'  ;
group by xxdm;
order by xxdm ASC;
INTO TABLE  县理应模拟录取总分上线人数.dbf
copy to  县理应模拟录取总分上线人数.xls  type  xl5
```

（9）全市各县区理应模拟录取上线人数：

```
Select  xqmc  as 县区名称 ,count(*) as  理应参考人数,;
```

（下转第 72 页）

# 基于.NET的学生信息管理系统

畅育超

**摘　要**：以学生信息管理系统为例，讲述在ASP.NET 2.0环境中采用B/S架构实现学生相关信息的查询、编辑、更新、汇总、删除等操作。

**关键词**：GridView控件；DetailsView控件；模板列；VB.NET+SQL Server 2000开发；ADO.NET编程

## 1　引言

随着信息技术的不断发展，特别是互联网技术的发展，为各类学校的学生信息管理提出了新的要求，学生信息处理、保存、查阅都要求及时、准确。本文就以基于.NET的学生信息管理系统为例，讲述在ASP.NET2.0环境中采用B/S架构实现学生相关信息的查询、编辑、更新、汇总、删除等操作和相关技术的实现。

## 2　学生信息管理系统的实现

### 2.1　登录页面

如图1所示。


**图1　登录页面设计**

（1）页面的主要功能。

在此页面中先选择“用户类型”，用户类型有两种：一是管理员，一是学生。学生以自己的学号和密码（出生日期的前6位）进行登录；管理员以自己的用户名和相对应的密码进行登录，只要用户名密码正确，即可进入后台管理页面。

（2）页面主要代码：

```
Protected Sub Button1_Click (ByVal sender As Object, ByVal e As EventArgs) Handles Button1.Click ' 登录事件
    If DownList_lx.Text.ToString() = "管理员" Then
        Try
            Using sqlconn As New SqlConnection()
            sqlconn.ConnectionString = ConfigurationManager.ConnectionStrings("sqlConnectionString").ConnectionString
            sqlconn.Open()
            Dim strSQL As String = "Select * From pass where username='" + admin_name.Text.ToString() + "' "
            Dim command As New SqlCommand(strSQL, sqlconn)
            Dim dataReader As SqlDataReader = command.ExecuteReader()
            If dataReader.Read() Then
                Dim mypassword As String = dataReader("userpassword").ToString()
                If mypassword = admin_pwd.Text.ToString() And DownList_lx.Text.ToString() = "管理员" Then
                    Session ("admin") = dataReader ("username").ToString()
                    Session ("loginpwd") = dataReader ("userpassword").ToString()
                    Response.Redirect("indexadmin.aspx") ' 系统后台管理页面
                Else
                    Response.Write (" <script language = 'javascript'>window.confirm(' 密码不正确，请后退重填！！ ');</script>")
                    Response.Write (" <script language = 'javascript'>parent.window.history.go(-1);</script>")
                End If
            Else
                Response.Write (" <script language = 'javascript'>window.confirm(' 用户名不正确！！ ');</script>")
                Response.Write (" <script language = 'javascript'>parent.window.history.go(-1);</script>")
            End If
            dataReader.Close()
            sqlconn.Close()
            sqlconn.Dispose()
        End Using
    Catch err As Exception
        Dim str As String = err.Message.Replace("'", "")
        Response.Write ("<script>alert ('" + str + "');</
```

```
script>") ′显示输错信息!
            End Try
        Else
            Try
                Using sqlconn As New SqlConnection()
            sqlconn.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
                    sqlconn.Open()
            Dim strSQL As String = "Select * From tablstudent
where sno='" + admin_name.Text.ToString() + "' "
                    Dim command As New SqlCommand(strSQL,
sqlconn)
                        Dim dataReader As SqlDataReader
= command.ExecuteReader()
                    If dataReader.Read() Then
                        Dim mypassword As String = Mid(Format
(dataReader ("brithday"), "yyyy-MM-dd"), 1, 4) & Mid(Format
(dataReader("brithday"), "yyyy-MM-dd"), 6, 2) ′把 1992-5-
′14 转换成 199205,6 位
                        If mypassword = admin_pwd.Text.ToString
() And DownList_lx.Text.ToString() = "学生" Then
                            Session ("admin") = dataReader ("sno").
ToString()
                            Response.Redirect("stubrows.aspx") ′学
′生学籍及成绩查询页面
                        Else
                            Response.Write (" <script language =
'javascript'>window.confirm(' 密码不正确,请后退重填! ');</
script>")
                        Response.Write (" <script language =
'javascript'>parent.window.history.go(-1);</script>")
                    End If
                Else
                    Response.Write (" <script language =
'javascript'>window.confirm(' 学号不正确,请后退重填! ');</
script>")
                    Response.Write (" <script language =
'javascript'>parent.window.history.go(-1);</script>")
                End If
                dataReader.Close()
                sqlconn.Close()
                sqlconn.Dispose()
            End Using
        Catch err As Exception
            Dim str As String = err.Message.Replace("'", "")
    Response.Write("<script>alert('" + str + "');</script>")
        End Try
    End If
End Sub
```

## 2.2 后台管理页面

当在图 1 中输入正确的管理员名称和密码，即可进入后台管理页面，本页面采用框架结构，能比较好地实现信息的集中管理，管理员只需在页面左侧的树形导航栏单击，即可进行相关信息的处理。如图 2 所示。


图 2 后台管理页面设计

## 2.3 学生基本信息页面

在本系统中，作为基础数据的学生基本信息输入是不可或缺的，为了使输入数据的准确性、一致性，在学生的基本信息中有些字段采用 DropDownList 提供数据源比较方便。具体的效果如图 3 所示。


图 3 学生基本信息页面设计

（1）页面的主要功能。

此页面主要完成学生基本信息的输入和相关信息的验证。基本信息中的学号、入学年度由系统自动生成，而且是只读的，这样就能避免学号混乱重复现象的发生。

（2）页面主要代码：

```
Protected Sub stuaddbuttom_Click (ByVal sender As Object,
ByVal e As System.EventArgs) Handles stuaddbuttom.Click
′添加学生信息事件
    Dim sqlcn As New SqlConnection()
        sqlcn.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
    sqlcn.Open()
     If Text_stuname.Text = "" Or sfen_id.Text = "" Or
Text_brith.Text = "" Or Text_address.Text = "" Then
        Response.Write (" <script language ='javascript' >
window.confirm(' 姓名、入学年度、出生日期、家庭地址等不能
为空,请后退重填!! ');</script>")
        Response.Write (" <script language ='javascript' >
parent.window.history.go(-1);</script>")
        Text_stuname.Focus()
        Exit Sub
```

```
Else
        Dim strSQL As String = "insert into tablstudent(sno,
sname,sex,brithday,nation,classname,class_bj,Identity_id,
address,memory,schoolyear,schoolmonth,stuxl,stustate)
values('" & Text_sno.Text.Trim() & "','" & Text_stuname.Text.
Trim() & "','" & Drop_sex.Text.Trim() & "','" & Text_brith.Text.
Trim () & "','" & Drop_nation.Text.Trim () & "','" & Drop_class.
Text.Trim () & "','" & DD_BJ_1.Text.Trim () & "','" & sfen_id.
Text.Trim() & "','" & Text_address.Text.Trim().ToString & "' ,'"
& Text_memory.Text.Trim () & "','" & Text_year.Text.Trim().
ToString & "','" & Drop_jjtt.Text.Trim () & "','" &
DropDownList1.Text.Trim() & "','" & Drop_state.Text.Trim() &
"')"
            Dim command As New SqlCommand(strSQL, sqlcn)
            command.ExecuteNonQuery()
            sqlcn.Close()
            Text_stuname.Text = ""
            Text_brith.Text = ""
            Text_address.Text = ""
            sfen_id.Text = ""
            Text_memory.Text = ""
        End If
        snonumber() ' 调用自动编号
    End Sub
Public Sub snonumber() ' 学号自动编号模块
        Dim sqlcon As New SqlConnection()
                                    sqlcon.ConnectionString =
ConfigurationManager . ConnectionStrings ( " sqlConnectionStr
ing").ConnectionString
        sqlcon.Open()
        Dim strsql As String
         strsql = "select sno from tablstudent order by sno
desc"
        Dim cmd As New SqlCommand(strsql, sqlcon)
                        Dim kcreader As SqlDataReader =
cmd.ExecuteReader()
        Dim p As Integer
        If kcreader.Read() Then
            p = Val(Right(RTrim(kcreader("sno")), 4)) + 1 ' 因为"
sno"字段的长度为 9
            st = Format(p, "000000")
        Else
            st = "000001"
        End If
        Text_sno.Text = DateTime.Today.Year().ToString & st
' 生成学号
        kcreader.Close()
        sqlcon.Close()
    End Sub
```

## 2.4 学生信息查询及编辑页面

如图 4 所示。


图 4 学生信息查询及编辑页面设计

(1) 页面的主要功能：

本页面采用查询的方式，将查询结果逐条呈现在 DetilsView 控件中便于对学生信息进行编辑、更新、删除等操作。

(2) 页面主要代码：

```
Protected Su b Button_cx_Click (ByVal sender As Object,
ByVal e As System.EventArgs) Handles Button_cx.Click ' 查
' 询按钮事件
        r = Text_tj.Text.ToString.Trim
        s = DDList_1.Text.ToString.Trim
        serch_data(s, r)
    End Sub
Public Sub serch_data(ByVal s As String, ByVal r As String)
' 查询函数模块
        If Text_tj.Text.ToString.Trim("") = "" Then
            Response.Write  (" <script  language ='javascript' >
window.confirm(' 查询内容不能为空！ ');</script>")
            Text_tj.Focus()
            Exit Sub
        End If
        Select Case DDList_1.Text.ToString.Trim("")
            Case "学号"
                s = "sno"
            Case "姓名"
                s = "sname"
            Case "班级名称"
                s = "class_bj"
            Case "入学年度"
                s = "schoolyear"
        End Select
        Dim sqlcn As New SqlConnection
            sqlcn.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
        sqlcn.Open()
            Dim strSQL As String = "Select sno,sname,sex,
brithday,nation,schoolyear,classname,class_bj,identity_id,
address,schoolmonth,stuxl,stustate,memory From tablstud
ent where " + s + " = '" + r + "' "
        Dim da As New SqlDataAdapter(strSQL, sqlcn)
        Dim ds As New DataSet()
        da.Fill(ds, "tablstudent")
        Dim dv As DataView
```

```
    dv = ds.Tables(0).DefaultView
    If dv.Count <= 0 Then ′判断记录数量
        Text_tj.Text = ""
        Text_tj.Focus()
         Response.Write  (" <script  language =′javascript′ >
window.confirm(′没有符合条件的记录！ ′);</script>")
    Else
        Details_1.DataSource = dv
        Details_1.DataBind()
        da.Dispose()
        ds.Dispose()
        sqlcn.Close()
        sqlcn.Dispose()
    End If
  End Sub
Protected Sub Button_edit_Click (ByVal sender As Object,
ByVal e As System.EventArgs) ′设置 Detailsview 控件为编辑
′模式事件
      Details_1.ChangeMode(DetailsViewMode.Edit) ′设置
′Detailsview 控件为编辑模式
      Button_cx_Click(Nothing, Nothing) ′调用按钮的单击事
′件！切记！
 Drop_classdata() ′调用 dropdownlist 控件动态添加数据函数
  End Su
Protected Sub Button_del_Click (ByVal sender As Object,
ByVal e As System.EventArgs) ′删除按钮事件
    Dim snoid As String
    snoid = Details_1.DataKey.Value.ToString.Trim ′获得当
′前记录的字段(学号)的值
    Dim result As String
    result = MsgBox("确定要删除此学生信息吗？ ", 4 + 48,
"信息提示！ ")
    If (result = vbYes) Then
        delete_stusno_data(snoid)
        Button_cx_Click(Nothing, Nothing) ′调用按钮的单击
′事件！切记！
    Else
        Response.Write  (" <script  language =′javascript′ >
parent.window.history.go(−1);</script>")
    End If
  End Sub
Public Sub delete_stusno_data(ByVal stu_id As String) ′删除
′数据函数模块
    Dim sqlcon As New SqlConnection()
      sqlcon.ConnectionString  =  ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
    Dim strSQL As String
    sqlcon.Open()
strSQL = "delete  from  tablstudent  where sno=" & stu_id & ""
    Dim command As New SqlCommand(strSQL, sqlcon)
    command.ExecuteNonQuery()
    sqlcon.Close()
  End Sub
Protected Sub Button_updata_Click(ByVal sender As Object,
ByVal e As System.EventArgs) ′更新数据
    Dim sqlcon As New SqlConnection()
        sqlcon.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
    Dim strSQL As String
    sqlcon.Open()
     Dim  stu_name,  stu_sfz,  stu_add,  stu_tel,  stu_bz,
stu_br, stu_snoid As String
stu_snoid = Details_1.DataKey.Value.ToString.Trim ′取出学号
       Details_1.Rows  (1).Cells  (1).FindControl  ("
TextBox_stuname").Focus() ′设置编辑焦点
     stu_name  =  CType  (Details_1.Rows  (1).Cells  (1).
FindControl  ("TextBox_stuname"),  TextBox).Text.ToString ().
Trim() ′取出姓名
  stu_sfz  =  CType (Details_1.Rows (8).Cells (1).FindControl ("
TextBox_sfz"), TextBox).Text.ToString().Trim() ′取出身份证号
    stu_add = CType(Details_1.Rows(9).Cells(1).FindControl
("TextBox_addres"), TextBox).Text.ToString().Trim() ′取出地址
  stu_tel  =  CType (Details_1.Rows (10).Cells (1).FindControl ("
TextBox_tel"), TextBox).Text.ToString().Trim() ′取出电话
    stu_bz = CType(Details_1.Rows(13).Cells(1).FindControl
("TextBox_bz"), TextBox).Text.ToString().Trim() ′取出备注
    stu_br = CType(Details_1.Rows(3).Cells(1).FindControl("
TextBox_brday"), TextBox).Text.ToString().Trim() ′取出生日
     sp  =  Details_1.Rows  (6).Cells  (1).FindControl  ("
DropDownList3") ′正确获取 Detailsview 控件列模板中的
′DropDownList 控件标示符(专业)
     st  =  Details_1.Rows  (7).Cells  (1).FindControl  ("
DropDownList4") ′正确获取 Detailsview 控件列模板中的
′DropDownList 控件标示符(班级)
    xl_down = Details_1.Rows  (11).Cells  (1).FindControl("
DrList2") ′获得 DropDownList 控件标示符(学历)
    zt_down = Details_1.Rows  (12).Cells  (1).FindControl("
DropDownList2") ′获得 DropDownList 控件标示符(状态)
    xb_down = Details_1.Rows  (2).Cells  (1).FindControl("
DropDownList1") ′获得性别值
    mz_down = Details_1.Rows  (4).Cells  (1).FindControl("
DropDList2") ′获得民族值
    If stu_name.Trim = "" Or stu_sfz.Trim = "" Or stu_add.
Trim = "" Then
  Response.Write  (" <script  language =′javascript′ >window.
confirm(′姓名、身份证号、地址不能为空,请确认！ ′);</script>")
    Else
   strSQL = "update tablstudent set sname=′" & stu_name
& "′,sex=′" & xb_down.Text.Trim & "′,brithday=′" & stu_br & "
′,nation =′" & mz_down.Text.Trim  & "′,classname =′" & sp.
Text.Trim & "′ , class_bj=′" & st.Text.Trim & "′, identity_id=′"
& stu_sfz & "′,address =′" & stu_add & "′,schoolmonth =′" &
stu_tel & "′,stuxl =′" & xl_down.Text.Trim & "′,stustate =′" &
zt_down.Text.Trim & "′,memory=′" & stu_bz & "′  where
```

```
sno='" & stu_snoid & "'"
    Dim command As New SqlCommand(strSQL, sqlcon)
          command.ExecuteNonQuery()
       End If
       sqlcon.Close()
    Button_cance_Click(Nothing, Nothing) ' 调用取消模块
    End Sub
```

## 2.5 学生退选课管理页面

(1) 页面主要功能。

本页面主要完成由管理员进行选课或退选的功能，它可完成某学年某学期某专业全部学生课程的选择或退选。为下一步学生成绩的管理提供了坚实的基础。效果如图 5 所示。

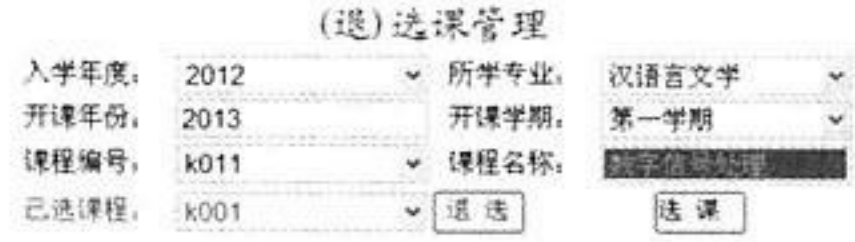



**图 5 学生退选课管理页面设计**

(2) 页面主要代码：

```
Protected Sub B_select_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles B_select.Click ' 提交本专业
' 全部学生本学期的全部学生的选课信息!
        If Text_year.Text < Trim (DDownList_nd.Text) Or Text_year.Text = "" Then
            Response.Write (" <script language ='javascript' > window.confirm (' 开课年份应小于等于入学年度或不为空,请重填!! ');</script>")
            Response.Write (" <script language ='javascript' > parent.window.history.go(-1);</script>")
        Else
            Dim sqlcon As New SqlConnection()
            sqlcon.ConnectionString = ConfigurationManager.ConnectionStrings("sqlConnectionString").ConnectionString
            sqlcon.Open()
            Dim strsql_sno As String
            strsql_sno = "select sno from tablstudent where schoolyear ='" & DDownList_nd.Text.Trim ("") & " ' and classname='" & DDownList_zy.Text.Trim("") & "'"
            Dim da As New SqlDataAdapter(strsql_sno, sqlcon)
            Dim ds As New DataSet()
            da.Fill(ds, "tablstudent")
            Dim i As Integer
    Dim cmd_insert As New SqlCommand(strsql_sno, sqlcon)
            For i = 0 To ds.Tables(0).Rows.Count - 1
                B_select.Enabled = False
        cmd_insert.CommandText = "insert into tablscore (sno, kkyear,kkday,kcn,cname,score,rscore) values ('" & Trim (ds.Tables(0).Rows(i).Item(0)) & " ','" & Text_year.Text.Trim("") & " ','" & Trim (DDownList_kkxq.Text) & " ','" & Trim (DDownList_kcbh.Text) & " ','" & Trim (text_kcmc.Text) & " ', null,null) "
                cmd_insert.ExecuteNonQuery()
            Next
            B_select.Enabled = True
            da.Dispose()
            ds.Dispose()
            sqlcon.Close()
        End If
        kcbhmc_add() ' 添加已选课程
    End Sub
Protected Sub B_delete_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles B_delete.Click ' 退选课程
        Dim strbh As String
        strbh = Trim(Drop_bhmc.Text.ToString) ' 取出课程编号
        kcbh_del(strbh)
         Response.Write (" <script language ='javascript' > window.confirm(' 编号为：" & Trim(Drop_bhmc.Text) & "的课程已被退选！ ');</script>")
        Response.Write ("<script language='javascript'>parent.window.history.go(-1);</script>")
    End Sub
Public Sub kcbh_del(ByVal strbh As String) ' 删除所选课程函数
        Dim sqlcon As New SqlConnection()
            sqlcon.ConnectionString = ConfigurationManager.ConnectionStrings("sqlConnectionString").ConnectionString
        sqlcon.Open()
        Dim strsql_bhmc As String
        strsql_bhmc = "delete from tablscore where kcn='" & strbh & " '"
    Dim cmd_bhmc As New SqlCommand(strsql_bhmc, sqlcon)
        Dim kccx_bhmc As SqlDataReader = cmd_bhmc.ExecuteReader()
        kccx_bhmc.Close()
        sqlcon.Close()
    End Sub
```

## 2.6 学生成绩管理页面

(1) 页面主要功能。

本页面根据学号查询完成学生成绩的管理，包括成绩的输入、修改、浏览的功能，效果如图 6 所示。

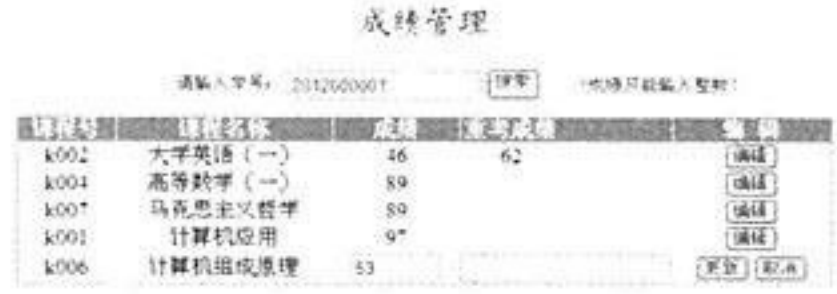



**图 6 学生成绩管理页面设计**

(2) 页面主要代码：

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click ' 查询按钮事件
        ksno = Trim(Text2_sno.Text)
        If ksno = "" Then
```

```
        Response.Write  (" <script  language ='javascript' >
window.confirm(' 学号不能为空!! ');</script>")
        Text2_sno.Focus()
      Else
        getdata(ksno)
      End If
    End Sub
Public Sub getdata(ByVal ksno As String) ' 数据浏览模块
      Dim sqlcn As New SqlConnection()
          sqlcn.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
      sqlcn.Open()
       Dim strSQL As String = "Select kcn,cname,score,
rscore From tablscore where sno='" & ksno & "'"
      Dim da As New SqlDataAdapter(strSQL, sqlcn)
      Dim ds As New DataSet()
      da.Fill(ds)
      GridView_2.DataSource = ds.Tables(0).DefaultView
      GridView_2.DataBind()
      da.Dispose()
      ds.Dispose()
      sqlcn.Close()
      sqlcn.Dispose()
    End Sub
Protected Sub gridview_2_RowUpdating (ByVal sender As
Object, ByVal e As System.Web.UI.WebControls.GridView
UpdateEventArgs) Handles GridView_2.RowUpdating ' 更新
数据事件
      Dim cusid, scoreid, rscoreid As String
      cusid = Trim (GridView_2.DataKeys (e.RowIndex) (0).
ToString) ' 取出课程编号
      scoreid = CType (GridView_2.Rows(e.RowIndex).Cells
(2).Controls(0), TextBox).Text.ToString ' 取出成绩
      rscoreid = CType (GridView_2.Rows(e.RowIndex).Cells
(3).Controls(0), TextBox).Text.ToString().Trim() ' 取出重考成绩
      updataview(cusid, scoreid, rscoreid) ' 调用更新函数
      GridView_2.EditIndex = -1
      updatlz() ' 零值更新模块
  Button1_Click(Nothing, Nothing) ' 调用命令按钮单击事件
    End Sub
```

## 2.7 班级管理页面

(1) 页面的主要功能。

实现班级信息的输入、修改、更新、浏览、删除的操作。效果如图 7 所示。


图 7 班级管理页面设计

(2) 页面主要代码:

```
Protected Sub Button_bj_Click (ByVal sender As Object,
ByVal e As System.EventArgs) Handles Button_bj.Click ' 添
' 加班级信息
      Dim sqlrs As New SqlConnection()
        sqlrs.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
      sqlrs.Open()
      If Textb_name.Text = "" Or Text_m.Text = "" Then
        Response.Write  (" <script  language ='javascript' >
window.confirm(' 班级名称、或班主任姓名不能为空,请后退重
填!! ');</script>")
        Response.Write  (" <script  language ='javascript' >
parent.window.history.go(-1);</script>")
        Exit Sub
      Else
        Dim strSQL As String = "insert into tablclass
(classno,classname,mangager) values('" & Textb_id.Text.Trim
("") & "','" & Textb_name.Text.Trim ("") & "','" & Text_m.Text.
Trim("") & "')"
        Dim command As New SqlCommand(strSQL, sqlrs)
        command.ExecuteNonQuery()
        sqlrs.Close()
        kcgetdata()
        Textb_name.Text = ""
        Textb_name.Focus()
      End If
      autonumber() ' 添加完课程调用自动编号模块
    End Sub
Public Sub kcgetdata() ' 数据浏览模块
      Dim sqlcn As New SqlConnection()
        sqlcn.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
      sqlcn.Open()
      Dim strSQL As String = "Select * From tablclass"
      Dim da As New SqlDataAdapter(strSQL, sqlcn)
      Dim ds As New DataSet()
      da.Fill(ds)
      GridView_3.DataSource = ds.Tables(0).DefaultView
      GridView_3.DataBind()
      da.Dispose()
      ds.Dispose()
      sqlcn.Close()
      sqlcn.Dispose()
    End Sub
Public Sub deleteclassdata(ByVal cusid As String) ' 删除数据
' 模块
      Dim sqlcon As New SqlConnection()
          sqlcon.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
      Dim strSQL As String
      sqlcon.Open()
```

```
        strSQL = "delete from tablclass where classno='" &
cusid & "'"
        Dim command As New SqlCommand(strSQL, sqlcon)
        command.ExecuteNonQuery()
        sqlcon.Close()
         End Sub
```

## 2.8 课程管理页面

（1）页面主要功能。

本页面主要完成课程相关信息的输入、修改、浏览、删除等操作。效果如图 8 所示。


**图 8 课程管理页面设计**

（2）页面主要代码：

```
Protected Sub buttom_kc_Click (ByVal sender As Object,
ByVal e As System.EventArgs) Handles buttom_kc.Click ' 添
' 加课程事件
        Dim sqlrs As New SqlConnection()
          sqlrs.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
        sqlrs.Open()
         If Text1_kcn.Text = "" Or Text2_kcm.Text = "" Or
dorp_kc.Text = "" Or Text_xf.Text = "" Then
            Response.Write ("<script language ='javascript' >
window.confirm(' 课程号、课程名、课程类型或学分不能为空，
请重填！！ ');</script>")
            Response.Write ("<script language ='javascript' >
parent.window.history.go(-1);</script>")
            Exit Sub
        Else
            Dim strSQL As String = "insert into tablcourse(cno,
cname,cnxf,cntype) values('" & Text1_kcn.Text.Trim() & "','" &
Text2_kcm.Text.Trim () & "','" & Text_xf.Text.Trim () & "','" &
dorp_kc.Text.Trim() & "' )"
            Dim command As New SqlCommand(strSQL, sqlrs)
            command.ExecuteNonQuery()
            sqlrs.Close()
            kcgetdata()
            Text2_kcm.Text = ""
        End If
        autonumber() ' 添加完课程调用自动编号模块
       End Sub
Protected Sub GridView_2_RowUpdating (ByVal sender As
Object, ByVal e As System.Web.UI.WebControls.GridViewUp
dateEventArgs) Handles GridView_2.RowUpdating ' 数据更新
        Dim cusid, cusname, custype, cuxf As String
        cusid = Trim (GridView_2.DataKeys (e.RowIndex) (0).
ToString) ' 取出课程编号
        cusname = CType(GridView_2.Rows(e.RowIndex).Cells
(1).Controls(0), TextBox).Text.ToString().Trim() ' 取出课程名称
        cuxf = CType (GridView_2.Rows (e.RowIndex).Cells (2).
Controls(0), TextBox).Text.ToString().Trim() ' 取出学分
        custype = CType(GridView_2.Rows(e.RowIndex).Cells(3).
FindControl ("DropDownList1"), DropDownList).SelectedItem.
Text ' 取出课程类型
  updataview(cusid, cusname, cuxf, custype) ' 调用更新函数
        GridView_2.EditIndex = -1
        kcgetdata()
    End Sub
Protected Sub GridView_2_RowEditing (ByVal sender As
Object, ByVal e As System.Web.UI.WebControls.GridView
EditEventArgs) Handles GridView_2.RowEditing ' 选定当前编
' 辑行
        GridView_2.EditIndex = e.NewEditIndex
        kcgetdata()
            ' 把数据源中当前记录的课程类型显示在模板列中的
'DropDownList1 控件中
        Dim sp As New DropDownList
        Dim hfd_1 As New HiddenField ' 隐藏字段
        Dim i As Integer
         sp = GridView_2.Rows (e.NewEditIndex).Cells (3).
FindControl("DropDownList1")
         hfd_1 = GridView_2.Rows (e.NewEditIndex).Cells (3).
FindControl("hfd_ktype")
        For i = 0 To sp.Items.Count - 1
           If sp.Items(i).Value.Trim = hfd_1.Value.Trim Then
              sp.Items(i).Selected = True
              Exit For
           Else
              sp.Items(i).Selected = False
           End If
        Next
            ' 把数据源中当前记录的课程类型显示在模板列中的
'DropDownList1 控件中
     End Sub
```

## 2.9 专业管理页面

（1）页面主要功能。

本页面主要实现专业信息的输入、修改、更新、浏览、删除等操作。效果如图 9 所示。


**图 9 专业管理页面设计**

(2) 页面主要代码：

```
Protected Sub buttom_zy_Click (ByVal sender As Object, ByVal e As System.EventArgs) Handles buttom_zy.Click ' 添
' 加专业事件
    Dim sqlcn As New SqlConnection()
        sqlcn.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
    sqlcn.Open()
    If text_id.Text = "" Or Text_name.Text = "" Or text_x.
Text = "" Then
        Response.Write ("<script language ='javascript' >
window.confirm(' 专业号或专业名不能为空,请后退重填!! ');<
/script>")
        Response.Write ("<script language ='javascript' >
parent.window.history.go(-1);</script>")
        Exit Sub
    Else
        Dim strSQL As String = "insert into tablspec
(specialityno,specialityname,spectype,specyear) values ('" &
text_id.Text & "','" & Text_name.Text & "','" & droplist_1.Text
& "','" & text_x.Text & "' )"
        Dim command As New SqlCommand(strSQL, sqlcn)
        command.ExecuteNonQuery()
        sqlcn.Close()
        getdata()
        autonumber() ' 调用自动编号模块
        Text_name.Text = ""
        Text_name.Focus() ' 设置输入焦点
    End If
End Sub
```

## 2.10 用户管理页面

(1) 页面的主要功能。

本页面的主要功能是完成系统用户的添加、删除等功能，效果如图 10 所示。

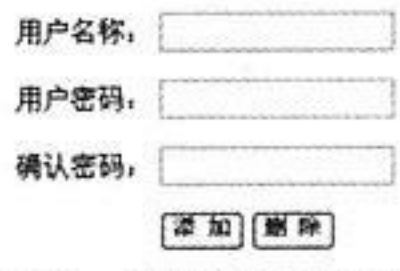



图 10 用户管理页面设计

(2) 页面主要代码：

```
Protected Sub Button_pass_Click (ByVal sender As Object, ByVal e As System.EventArgs) Handles Button_pass.Click
' 添加用户事件
    Dim sqlcon As New SqlConnection()
      sqlcon.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
    sqlcon.Open()
    Dim sqlstring As String
    Dim sqlstring1 As String
    If Trim (Text_pass1.Text) <> Trim (text_pass2.Text) Or
Text_name.Text = "" Then
        Response.Write ("<script language ='javascript' >
window.confirm (' 用户名不能为空或两次输入密码不一致,请
重填!! ');</script>")
        Response.Write ("<script language ='javascript' >
parent.window.history.go(-1);</script>")
    Else
        sqlstring1 = "select username from pass where
username='" & Trim(Text_name.Text) & "'"
    Dim cmd_add As New SqlCommand(sqlstring1, sqlcon)
        Dim dataReader As SqlDataReader = cmd_add.
ExecuteReader()
        If dataReader.Read() = True Then
            Response.Write ("<script language='javascript' >
window.confirm(' 用户名重复,请重填!! ');</script>")
            Response.Write ("<script language='javascript' >
parent.window.history.go(-1);</script>")
        Else
            dataReader.Close()
                sqlstring = "insert into pass (username,
userpassword) values ('" & Text_name.Text.Trim("") & " ','" &
Text_pass1.Text.Trim("") & " ') "
    Dim cmd_insert As New SqlCommand(sqlstring, sqlcon)
            cmd_insert.ExecuteNonQuery()
            sqlcon.Close()
        End If
    End If
    Text_name.Text = ""
End Sub
Protected Sub Button_del_Click (ByVal sender As Object,
ByVal e As System.EventArgs) Handles Button_del.Click ' 删
' 除用户事件
    Dim sqlcon As New SqlConnection()
      sqlcon.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
    sqlcon.Open()
    Dim sqlstring As String
    Dim sqlstring1 As String
 sqlstring1 = "select * from pass where username ='" &
Text_name.Text.Trim ("") & " ' and userpassword ='" &
Text_pass1.Text.Trim("") & " '"
    sqlstring = "delete from pass where username ='" &
Text_name.Text.Trim ("") & " ' and userpassword ='" &
Text_pass1.Text.Trim("") & " '"
    Dim cmd_sel As New SqlCommand(sqlstring1, sqlcon)
      Dim datasel As SqlDataReader = cmd_sel.
ExecuteReader()
    If datasel.Read() = False Then
        Response.Write ("<script language ='javascript' >
window.confirm(' 用户名不存在,请重填!! ');</script>")
```

```
        Response.Write  (" <script  language ='javascript' >
parent.window.history.go(-1);</script>")
    Else
        datasel.Close()
    Dim cmd_del As New SqlCommand(sqlstring, sqlcon)
        cmd_del.ExecuteNonQuery()
        Text_name.Text = ""
         Response.Write  (" <script  language ='javascript' >
window.confirm(' 删除成功！ ');</script>")
        Response.Write  (" <script  language ='javascript' >
parent.window.history.go(-1);</script>")
    End If
    sqlcon.Close()
End Sub
```

## 2.11 学生学籍及成绩查询页面

（1）页面的主要功能。

本页面的主要功能是学生利用学号及密码可以查询自己学籍、所开设课程及学习成绩。效果如图 11 所示。

学生 杨逸凡 的成绩如下：

学号：2012000001　姓名：杨逸凡　学历层次：本科　专业：计算机科学与技术
已完成总学分：13　总平均分：74

| 序号 | 开课年度 | 开课学期 | 课程编号 | 课程名称 | 学分 | 课程类型 | 成绩 | 查考成绩 |
|---|---|---|---|---|---|---|---|---|
| 1 | 2012 | 第一学期 | k002 | 大学英语（一） | 2 | 公共课 | 46 | 62 |
| 2 | 2012 | 第一学期 | k004 | 高等数学（一） | 3 | 公共课 | 89 | |
| 3 | 2012 | 第一学期 | k007 | 马克思主义哲学 | 2 | 公共课 | 89 | |
| 4 | 2012 | 第一学期 | k001 | 计算机应用 | 1 | 公共课 | 97 | |
| 5 | 2012 | 第一学期 | k006 | 计算机组成原理 | 5 | 专业课 | 53 | |

图 11 学生籍贯及成绩查询页面设计

（2）页面主要代码：

```
Protected Sub Page_Load (ByVal sender As Object, ByVal e
As System.EventArgs) Handles Me.Load ' 页面加载事件
    Dim sqlcon As New SqlConnection()
        sqlcon.ConnectionString = ConfigurationManager.
ConnectionStrings("sqlConnectionString").ConnectionString
    sqlcon.Open()
    Dim strsql As String
    Dim strsno As String
    Dim sqlstring As String
    strsno = Session("admin")
     strsql = "select * from tablstudent where sno ='" +
strsno + "'"
    Dim command As New SqlCommand(strsql, sqlcon)
        Dim dataReader As SqlDataReader = command.
ExecuteReader()
    If dataReader.Read() Then
        TextBox_name.Text = dataReader("sname").ToString
        TextBox_sno.Text = dataReader("sno").ToString
        TextBox1_name.Text = dataReader("sname")
        TextBox_xl.Text = dataReader("stuxl")
        TextBox_spc.Text = dataReader("classname")
        dataReader.Close()
    End If
     sqlstring = "select s.kkyear,s.kkday,s.kcn,s.cname,m.
cnxf,m.cntype,s.score,s.rscore from tablscore as s,tablcourse
as m where sno='" + strsno + "'and s.kcn=m.cno "
    Dim cmd_sqlreader As New SqlDataAdapter(sqlstring,
sqlcon)
    Dim readerdata As New DataSet
    cmd_sqlreader.Fill(readerdata)
    ' 统计成绩的平均值方法(一)
    Dim dt As DataTable ' 定义一个表，实质就是 Dataset 的
' 副本
    Dim dr As DataRow ' 定义数据行
    dt = readerdata.Tables(0)
    dr = readerdata.Tables(0).NewRow
    GridView_scoreBrows.DataSource = readerdata.Tables
(0).DefaultView
    Dim objSum As Object ' 定义对象存放学分的合计
    Dim objaverage As Object ' 定义对象存放成绩的平均值
    objSum = dt.Compute("Sum(cnxf)", "score>=0") ' 求考
' 试成绩大于等于 0 的课程学分的和
     objaverage = dt.Compute("avg(score)", "score>=0") ' 求
' 考试成绩的平均值
    Label_total.Text = objSum.ToString
    Label_average.Text = objaverage.ToString
    GridView_scoreBrows.DataBind()
    cmd_sqlreader.Dispose()
    readerdata.Dispose()
    sqlcon.Close()
    sqlcon.Dispose()
End Sub
  Protected Sub GridView_scoreBrows_RowDataBound
(ByVal sender As Object, ByVal e As System.Web.UI.
WebControls.GridViewRowEventArgs) Handles GridView_sc
oreBrows.RowDataBound
    Dim id As Int16
    If e.Row.RowIndex <> -1 Then
        id = e.Row.RowIndex + 1
        e.Row.Cells(0).Text = id.ToString()
    End If
End Sub
```

## 3 结语

较完整地讲述了采用 B/S 架构完成学生信息管理系统大部分功能模块，本系统能够较好地完成学生相关信息的处理，特别是学生能实时随地查询自己学籍及学习成绩的相关信息。本系统的源代码以源程序为准。

（收稿日期：2013-03-13）

# 基于 Web 的标准信息服务系统设计与实现

张 洪　张海静

**摘　要**：基于 Web 的标准信息服务系统运用 ASP 技术，以 SQL Server2005 为后台数据库，实现向企事业单位用户提供标准服务的目的。系统加大了标准资源更新和整合力度，强化了服务领域的延伸，增强了资源保障能力和公共服务能力，具有很好的实用价值。

**关键词**：标准检索；信息服务系统；ASP 技术

## 1　引言

随着社会发展，越来越多企业意识到要想在竞争中取得先机，必须实现标准化。但国内各类企业、机构，获得标准信息的渠道相对比较分散、狭窄，不能及时了解自身使用标准的变更情况，标准目录信息更新时，企业、机构得到的信息滞后，未能实现标准跟踪、预警功能，所使用的生产标准、检测标准落后。基于 Web 的标准信息服务系统可以更好地为广大企业、机构服务，给用户提供个性化的帮助。

## 2　系统设计

### 2.1　开发及运行环境

(1) 系统开发平台：DreamWeaver8.0，ASP 技术。

(2) 运行环境

硬件：用于运行系统的计算机需具有宽带上网（配有 10M/100M 以太网卡）、或 ADSL 上网（配 ADSL 网卡，利用电话线）的功能。

软件：系统运行操作系统可为 MicroSoft Operating System，主要软件为网页浏览器。建议用户使用屏幕大小为 1280*960 分辨率。

### 2.2　数据库

系统的数据库采用 SQL Server2005，主要包括以下数据表：国内标准目录表 tbl_gb_temp、国外标准目录表 tbl_fb_temp、标准跟踪表 tbl_cus_traceBZ、标准专库主表 tbl_speLib、标准专库细化表 tbl_speLibBZNo、用户表 tbl_bzzluser、标准新闻公告表 tbl_news。

### 2.3　功能模块

系统由前台和后台两大块组成，前台包括公共服务子系统、工业服务子系统、农业服务子系统、服务业服务子系统，后台功能包括标准目录检索、标准专库检索、标准目录跟踪、标准文本下载、标准动态信息，如图 1 所示。

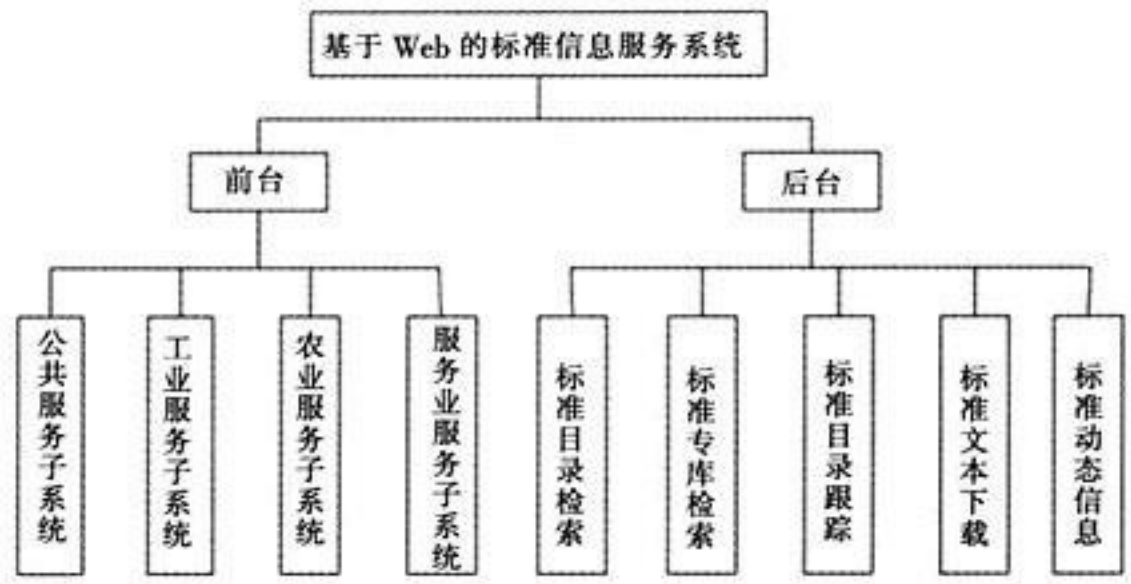



图 1　系统功能结构图

## 3　软件实现

### 3.1　前台引导页

引导页为用户提供不同子系统的入口，如图 2 所示。


图 2　系统引导页实现图

其中 WTO/TBT 图标提供另外一个系统的链接。

标准检索提供输入信息格式的提示，代码如下：

```
<input class = "text" type = "text" value = "" name = "keywords" / tips="     关键字输入说明:<BR>    1、输入标准号或标准名称关键字;<BR>    2、不必输入标准年份，增大标准检索范围。" onmousemove=seashowtip(this.tips,1,280) onmouseout=seashowtip(this.tips,0,280)>
```

### 3.2　公共服务子系统

该模块为用户提供标准综合信息的浏览，主要包括菜单导航条、用户登录板块、标准服务基地介绍板块、联系我们板块、标准检索板块、即将实施/即将作废/最新发布标准板块、标准新闻公告板块、标准专库分类板块。

### 3.3 工业服务子系统

该模块为工业类相关单位用户提供标准服务信息的浏览，主要包括菜单导航条、用户登录板块、标准检索板块、标准新闻公告板块、联系我们板块、工业综合类标准细化板块、电器电子类标准细化板块、通信类标准细化板块、仪器/仪表类标准细化板块、零部件类标准细化板块、化工材料类标准细化板块、金属材料类标准细化板块、机械装备类标准细化板块、消防类标准细化板块、节能减排类标准细化板块。

### 3.4 农业服务子系统

该模块为农业类相关单位用户提供标准服务信息的浏览，主要包括菜单导航条、用户登录板块、标准检索板块、农业地方标准板块、农业标准化知识板块、标准新闻公告板块、农业标准细化板块、林业标准细化板块、畜牧业标准细化板块、林业标准细化板块。

### 3.5 服务业服务子系统

该模块为服务业类相关单位用户提供标准服务信息的浏览，主要包括菜单导航条、用户登录板块、标准检索板块、服务业地方标准板块、服务业专题专栏板块、服务业综合类标准细化板块、服务标准细化板块、公共安全标准细化板块、职业健康标准细化板块、信息化标准细化板块、城市工程标准细化板块、交通与物流标准细化板块、环境管理标准细化板块。

### 3.6 后台登录

用户通过帐号、密码登录后台，登录信息使用 Session 进行保存，并进行日志记录。

### 3.7 标准目录检索模块

该模块主要实现标准目录的检索，可根据标准号、或中文标准名称、或英文标准名称、或中文主题词、或英文主题词、或中标分类号、或 ICS 分类号、或发布年份、或实施年份进行检索，可选择关键词模糊或精确匹配。ASP 检索主要代码如下：

```
sqlGB="select top 250 a100 as bzNo,a301 as bzName,1 as bzType,1 as canDown,bisdownfile,addDate,a200,a305 as page from tbl_gb_temp where 1=1"
    if selType="a825" then
        sqlFB = "select '' as bzNo,'' AS bzName,2 as bzType,0 as canDown,0 as bisdownfile,getdate() as addDate,'' as a200,0 where 1=0"
    else
 sqlFB="select top 250 a100 as bzNo,cast(a2981 AS nvarchar(300)) AS bzName,2 as bzType,1 as canDown,bisdownfile,addDate,'' as a200,page from tbl_fb_temp where 1=1"
    end if
    if selType="a100" then
        if likeType="%" then
            queryGB=" and a100 like '%"&strTitle&"%'"
            queryFB=" and a100 like '%"&strTitle&"%'"
        elseif likeType="=" then
            queryGB=" and a100 = '"&strTitle&"'"
            queryFB=" and a100 = '"&strTitle&"'"
        else
            queryGB=""
            queryFB=""
        end if
    elseif selType="a302" then
...
    else
        queryGB=""
        queryFB=""
    end if
    query =sqlGB&queryGB&" UNION "&sqlFB&queryFB&" order by a200 asc,bisdownfile desc"
```

### 3.8 标准专库检索模块

该模块主要实现标准专库的检索，检索条件同标准目录检索，ASP 检索代码中增加如下代码：

```
'speLib 为标准专库类别 ID
if speLib<>"" then
queryGB =queryGB&" and a100 in  (select speLibBZNo from tbl_speLibBZNo where speLibID='"&speLib&"') "
queryFB =queryFB&" and a100 in  (select speLibBZNo from tbl_speLibBZNo where speLibID='"&speLib&"') "
end if
```

### 3.9 标准目录跟踪模块

用户将检索到的标准添加到标准跟踪表，当该标准作废时，系统自动提醒该标准作废信息，并通过比对标准替代字段，找到替代该标准的新标准，提醒用户处理。主要实现代码如下：

```
<%
if (TBZIsOld="0") and (rs.fields("a200")="W" or rs.fields("a200")="L") then
        changebz=""
        locate1=InStr(rs.fields("a462"),"被")
        locate2=InStr(locate1,rs.fields("a462"),"代替")
        if locate2<1 then
            locate2=InStr(locate1,rs.fields("a462"),"替代")
        end if
        if locate1>0 and locate2>0 then
            changebz =mid  (rs.fields  ("a462"),locate1 +1, locate2-locate1-1)
        end if
    Set rs1 = Server.CreateObject("ADODB.Recordset")
        rs1.open "select * from tbl_gb_temp where a100='"&changebz&"'",conn,1,1
        if not rs1.eof then
            changebz=trim(rs1("a100"))
        else
            changebz=""
        end if
        rs1.close
```

```
        set rs1=nothing
        if changebz<>"" then
        '显示替代标准信息
%>
          …
<%
        end if
end if
%>
```

### 3.10 标准文本下载模块

该模块实现标准文本链接的下载：

```
response.write " <script language =javascript >location.replace("+rs.fields("bz_pdffilepath")+");</script>"
```

### 3.11 标准动态信息模块

标准动态信息包括即将实施标准、即将作废标准、最新发布标准。

即将实施标准显示最近 3 个月内即将实施的标准，主要实现代码如下：

```
sqlGB=sqlGB&" and CONVERT(varchar(12), a205, 112) >=CONVERT (varchar (12), CAST ("&startTime&"' AS smalldatetime), 112)"
sqlFB=sqlFB&" and CONVERT (varchar (12), a205, 112) >=CONVERT (varchar (12), CAST ("&startTime&"' AS smalldatetime), 112)"
sqlGB=sqlGB&" and CONVERT(varchar(12), a205, 112) <=CONVERT (varchar (12), CAST ("&endTime&"' AS smalldatetime), 112)"
sqlFB=sqlFB&" and CONVERT (varchar (12), a205, 112) <=CONVERT (varchar (12), CAST ("&endTime&"' AS smalldatetime), 112)"
```

最新发布标准显示最近 3 个月内新发布的标准，主要实现代码如下：

```
sqlGB=sqlGB&" and CONVERT(varchar(12), a210, 112) >=CONVERT(varchar(12), CAST('"&startTime&"' AS smalldatetime), 112)"
sqlGB=sqlGB&" and CONVERT(varchar(12), a210, 112) <=CONVERT (varchar (12), CAST ("&endTime&"' AS smalldatetime), 112)"
```

即将作废标准显示最近 3 个月内即将作废的标准，主要实现代码如下：

```
sqlGB=sqlGB&" and CONVERT(varchar(12), a225, 112) >=CONVERT (varchar (12), CAST ("&startTime&"' AS smalldatetime), 112)"
sqlFB=sqlFB&" and CONVERT (varchar (12), a101, 112) >=CONVERT (varchar (12), CAST ("&startTime&"' AS smalldatetime), 112)"
sqlGB=sqlGB&" and CONVERT(varchar(12), a225, 112) <=CONVERT (varchar (12), CAST ("&endTime&"' AS smalldatetime), 112)"
sqlFB=sqlFB&" and CONVERT (varchar (12), a101, 112) <=CONVERT (varchar (12), CAST ("&endTime&"' AS smalldatetime), 112)"
```

## 4 结语

本系统目前已在实际工作中得到推广运行，系统用户只需鼠标轻点系统，短短几分钟内就可知道标准是否作废、是否有新的相关标准发行，标准化工作效率将得到显著提升。系统的标准检索、过期预警、远程订购等模块具有领先的技术，强大的功能和简单方便的操作受到用户广泛好评。

## 参考文献

[1] 吴代文. 基于 Web 的资源共享网站的设计与实现. 渭南师范学院学报，2013，28（2）：77-80.
[2] 葛郁葱. 标准文献的特点及其检索方法. 情报杂志，2009，28（B12）：166-167.
[3] 郑阿奇. SQL Server 教程. 北京：清华大学出版社，2005.
（收稿日期：2013-04-22）

（上接第 34 页）

```
  Set Pointer = clper
End Property
```

图 2 Basic 实现链表数据结构测试程序运行结果

为了测试上述代码的正确性，编写一个小的测试程序，将 1，2，3，4，5，6，7，8，9，10 插入到链表中，并进行其基本的添加、删除操作。测试程序运行结果如图 2 所示。

从测试程序运行结果看：上述代码是成功的，测序程序能很好地完成链表的基本操作。

## 4 结语

尽管 Basic 语言没有指针类型数据，但是借鉴面向对象的程序设计思想，也可以实现诸如链表一类需要用到指针描述的数据结构的。当然，以上仅仅是举了单链表这个最简单的例子，更复杂的操作（如指定结点的插入与删除）还有待进一步的完善，读者可以作更深入的尝试，这对提高自己的对高级程序语言的理解是大有帮助的。

（收稿日期：2013-05-03）

# 基于 WAMP 的在线评测系统设计与实现

李臣龙　鲍广喜

**摘　要**：详细阐述了在线评测系统的结构原理，运用 PHP、Mysql 和 Apache 组合，设计实现了基于 B/S 结构的在线评测系统。实现了用户在线评测、查看状态、讨论等功能。

**关键词**：ACM/ICPC 竞赛；Online Judge 系统；PHP 语言；Mysql 数据库

## 1　引言

在线评测系统源于 ACM/ICPC（国际大学生程序设计竞赛）是由 ACM (Association for Computing Machinery，美国计算机协会) 组织的年度性竞赛，始于 1970 年，是全球大学生计算机程序能力竞赛活动中最有影响的一项赛事。

在线评测系统能够实时对用户提交的程序进行评测，对提高同学编程能力有很大帮助，同时也解决手工评测效率的的问题。本设计旨在提供一个训练的平台，也为希望提高计算机程序设计能力的同学提供一个练习的场所。

## 2　开发工具

“针对 ACM/ICPC 的在线评测系统”是基于 Internet 的 B/S 模式的。客户端可以直接在浏览器中粘贴相关题目的源代码，服务器接受到源代码之后将记录存入数据库，然后编译，执行，并验证源代码的正确性。服务器端是采用的是 Apache+ PHP+ Mysql 的组合方案。下面对用到的相关工具做简要介绍。

### 2.1　PHP

系统采用跨平台的服务器端嵌入式语言 PHP (Hypertext Preprocessor) 作为主开发语言。这是因为尽管目前的网络数据库开发工具众多 (例如：应用较广泛的有 ASP、PHP、JSP、CGI 等)，但 PHP 秉承了 Linux 的 GNU 风格，借助于公开的源代码，大量应用 C、C++、JAVA 的语法，具有易学、好用的优点。PHP 内置了文件上传密码认证、Cookie 操作、动态 GIF 生成、邮件收发、XML,共享内存等功能，而更为重要的是 PHP 可以直接连接多种数据库，并完全支持 ODBC,PHP 目前所支持的数据库有 Oricale，Mysql，Sybase，Informase/ Illustra 等。

### 2.2　MySQL 与 Apache

系统采用 Mysql 作为 SQL 数据库服务软件。这是因为 Mysql 具有多用户、多线程、速度快、易用性强的优点。Apache 虽与微软公司的 IIS 类似，同属于一种 Web 服务器，但 Apache 的优点使其成为已为目前最流行的 Web 服务器，故系统采用 Apache 作为 Web 服务器。

## 3　系统原理

系统采用流行的 B/S 模式。只要在客户端通过 IE 浏览器便可以访问和管理，大量的操作放在服务器端，包括请求反馈、数据存取、评判结果生成等等，只是将用户需要的结果通过 HTML 文档页面显示在客户端。基本工作模式如图 1 所示。

学生通过浏览器向 Web 服务器发出请求，服务器读取相应 PHP 文件，使用脚本解释引擎 PHP 文件进行解释执行，对后台数据库执行 SQL 命令，调用符合用户的应用服务，然后将结果返回浏览器。B/S 结构充分利用了服务器资源，实现了开发环境与应用环境的分离，易于维护，信息共享度高。

管理员可以通过浏览器可以对用户信息、题库等进行管理，负责整个系统的数据操作和维护，这样安全性更高，能够实现较为复杂管理机制。


**图 1　系统的工作模式**

## 4　系统功能

本设计主体思想是为了方便用户，提供一个练习、交流并提高编程水平的平台。设置了有用户注册、题目浏览、提交、查看提交状态、查看排名、论坛等功能。

### 4.1　注册，登录与提交模块

#### 4.1.1 注册模块

该模块的功能是实现考生注册。学生输入想要注册的用户名和密码等信息,提交之后由系统调用服务器 Mysql 数据库判断该用户是否已经存在，如果不存在，就初始化 Session 环境变量注册 session 变量，同时把用户输入的用户名和密码写到 Mysql 数据库中。

#### 4.1.2 登录模块

登录模块采用多级用户权限管理，从管理效率，数据安全上考虑，本系统采用严格的权限分级制度，不同用户对应不同

活动权限，以确保数据安全。在 B/S 模式下根据用户 ID 赋予用户不同权限，例如学生登录后可以浏览题目，查看自己状态，提交题目，参加 bbs 讨论等。管理员可以对数据库和系统资源进行管理，包括学生权限和资料管理，题目更新管理，消息和平台内公共资源管理等。

根据登录信息判断其合法性，然后在数据库中查找与之相匹配的记录，进行身份判断，定义用户权限级别及初始化 Session 环境变量等。如果没有该用户则提示其进行注册。判断用户类型 (学生、管理员) 、根据用户类型重定向用户首页 (学生首页、管理员首页)。

#### 4.1.3 提交模块

用户注册登录后可以浏览题目进行程序提交,提交信息存入个人数据库表中，程序交给后台评测系统进行处理。

### 4.2 后台评测系统

服务器后台运行守候程序，每一个时间间隔查询一次数据库，如发现有提交纪录，则启动评判进程，进行相关题目的评判。每道题目均对应一个数据输入输出文件，让评判进程启动一个子进程编译运行用户提交上来的程序，当程序启动时计时，在时间允许范围之内，如果其得到了正确答案，则修改数据库，令其 status 为 Accepted，否则，如果答案错，或者表达错，运行错等流程图，如图 2 所示。

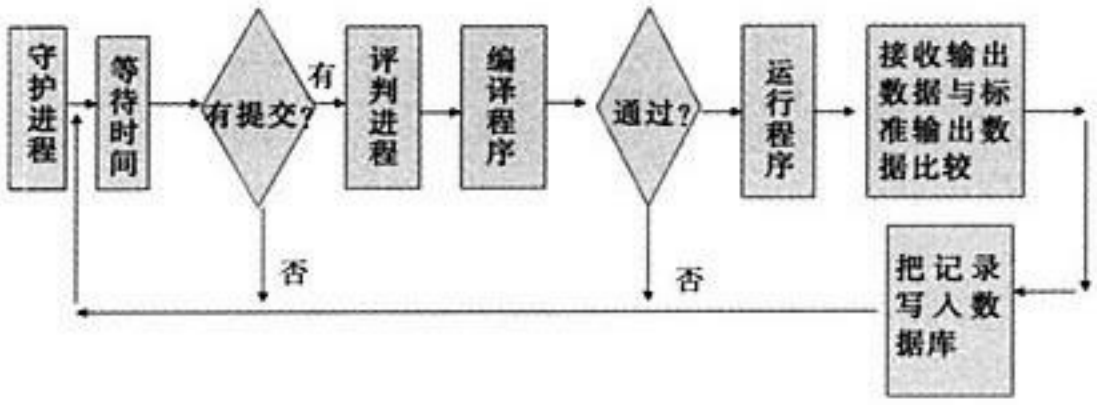



图 2 后台评测流程

### 4.3 关键技术

#### 4.3.1 程序编译

本设计实现了编译运行 C、C++、Java 程序的在线编译，下面以对 C/C++程序为例说明 Web 编译的实现。系统使用 VC 编译器，通过命令 cl 对代码进行编译运行，最后 PHP 调用外部命令 passthru 显示编译信息。

对提交程序转换为用户选择的语言格式保存到数据库，然后通过 PHP 调用 VC 的 cl 命令进行编译与运行：

```
<?
 include("topno.php");  include("runtime.php");
 global $language, $source;
if(strlen($source)<1)
{
//表单提交代码等信息
}
else{
if($language==4)// 针对某种语言产生编译文件
 {
 $file=fopen("data/$problem_id/$problem_id.cpp","w");
 fwrite($file,$source);
 fclose($file);
 }
  …//其他语言
}
chdir("data/$problem_id");
$command = "my1.bat $problem_id.cpp";
exec($command);//执行外部命令编译程序
global $file;
$dir=dir(".");
    while($file=$dir->read())//读取服务器端多个验证文件进行
//验证
{
if(strstr($file,".in"))
{
$command="$problem_id.exe <$file >datatemp.out";
$runtime = new Runtime(3);
passthru($command); //执行外部命令运行程序
echo $runtime->totaltime();
$dout=$file;  $dout=rtrim($dout,".in");
$dout=$dout.".out";
global $text1;global $text2;global $size1;global $size2;global
$dtemp;global $dsamp;
$arrtext1 = file("datatemp.out");  $arrtext2 = file("$dout");
  for ($i=0; $i < count($arrtext2); $i++)  //通过逐字符比较判断
//答案
{   $trimed1=rtrim($arrtext1[$i],"\n");//消除空格回车符的影响
    $trimed1=rtrim($trimed1,"\r");
    $trimed1=rtrim($trimed1," ");
    if($i! =(count($arrtext1)-1))
    $trimed1=$trimed1."\r"."\n";
    $trimed2=rtrim($arrtext2[$i],"\n");
    $trimed2=rtrim($trimed2,"\r");
    $trimed2=rtrim($trimed2," ");
    if($i! =(count($arrtext2)-1))
    $trimed2=$trimed2."\r"."\n";
     if($trimed1! =$trimed2)
     {
          //成功
     }
     else  continue;
     }
}}}
//其他处理…
?>
```

#### 4.3.2 程序评判

本设计采用逐字符比较的方式对程序进行评判。即评判程序每次从管道读入一个字符与标准输出数据比较，如果两个文件

(下转第 75 页)

# 解析 Website 与 WebApplication 的区别及相互转换

吕和乾

**摘　要**：分析了 ASP.NET 平台中两种 Web 开发的模式 Website 与 WebApplication 之间的区别，并通过源代码的分析，介绍了二者之间的相互转换。

**关键词**：ASP.NET 技术；Website 模式；WebApplication 模式

在 ASP.NET 系统的开发中经常会遇到两种 Web 开发的模式：Website 与 WebApplication 。那么怎样进行选择，如何进行取舍呢？下面就来分析一下这两种模式的区别，希望能够对大家有所帮助。

## 1　两种模式优缺点

Website 模式开发管理简单，拷贝即添加到项目中，没有命名空间的干扰，但不利于工程化开发，比如代码出错不容易发现。可以把一个目录当做一个 Web 应用来处理，是 VS2008 推荐的一种 Web 开发模式，如果网站简单，作业不是很多用 Website，适合开发中小网站。

WebApplication 使用到程序池，是 VS2003 提供的主要开发模式，后来在 VS2005 SP1、VS2008 中微软提供了这两种 Web 开发模式。因为 WebApplication 采用和 VS2003 一样的开发结构，所以可以方便升级。WebApplication 可以将网站拆分成多个项目以方便管理和团队开发，可以从项目中和源代码管理中排除一个文件，拥有更强大的代码检查功能。如果网站复杂，作业量庞大，建议使用这种模式，用 WebApplication 切割成多个 Project，这样编译起来也快一些。

## 2　两种模式文件结构

图 1 展示的是 Website 模式的文件机构：

图 1 中可以看出一个网页文件由两个文件组成，其中 TestWebsite.aspx 为前台页面文件，TestWebsite.aspx.cs 为后台代码文件。

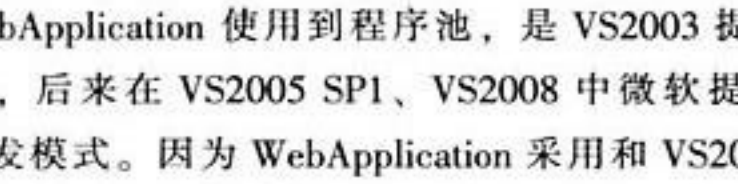



**图 1　Website 模式的文件结构**

图 2 展示的是 WebApplication 模式的文件结构：

图 2 中可以看到 WebApplication 的一个网页文件由 3 部分组成，其中 TestWebApplication.aspx 为页面文件，TestWebApplication.aspx.cs 为后台代码文件，TestWebApplication.aspx.designer.cs 是窗体设计器生成的代码文件，作用是对窗体上的控件做初始化工作。


**图 2　WebApplication 模式的文件结构**

这只是文件外观上的区别，当然文件的内部结构也有所不同，图 3 和图 4 是 Website 模式的文件内的结构。

**图 3　Default.aspx 文件代码**

**图 4　Default.aspx.cs 文件代码**

图 5 和图 6 是 WebApplication 模式的文件内部结构。

**图 5　Default.aspx 文件代码**

图 6　Default.aspx.cs 文件代码

注意图片标框部分，由此可以清楚地看到，Website 模式使用的是 Codefile，没有使用命名空间；Web Application 模式使用的是 CodeBehind 和相应的 namespace 空间。

## 3　模式转换

为什么要了解上面所讨论的这些不同呢？因为根本目的就是通过上面的学习，实现二者之间的相互转换。做开发时经常要移植到其他不同模式的开发环境中，为了节省时间快速开发，就必须要掌握它们之间的相互转换。

（1）WebApplication 转 Website，网上暂时未查到专门的转换工具，但通过比较后发现如果转换也非常简单，删除所有 *.designer.cs，再将 *.aspx、*.ascx、*.master 等页面文件中的 Codebehind=" *.aspx.cs" 批量替换成 CodeFile=" *.aspx.cs"，然后将这些文件拷贝到 Website 模式的应用系统下，这样就实现了转换。

例如将图 2 所示的 WebApplication 中的TestWebApplication.aspx 网页文件成功迁移到 Website 中，并且成功进行了运行测试。如图 7 所示。

图 7　运行测试

通过查看 TestWebApplication.aspx.cs 文件的原代码会发现代码中包含“namespace WebApplication”命名空间，不过不影响运行。因为 Website 模式缺省是不使用命名空间的，为了一致可以一并删除，如图 8 所示。

（2）Web 网站转换成 Web 应用程序相对要简单些，将 *.aspx 和 *.aspx.cs 文件拷贝到 WebApplication 中，然后在“解决方案资源管理器”中进行右键转换即可，如图 9 所示。

将图 1 所示的 Website 中的 TestWebsite.aspx 移植到 WebApplication 中进行测试，操作方法是先在文件夹中选择“TestWebsite.aspx”、“TestWebsite.aspx.cs”这两个文件，然后到 WebApplication 的“解决方案资源管理器”中右键进行“粘贴”，然后进行转换，会发现转换后会自动生成“TestWebsite.aspx.designer.cs”文件，并且 TestWebsite.aspx 文件代码中的 CodeFile 自动变成了 Codebehind，测试结果是转换成功，能够正常运行。如图 10 所示。

图 8　TestWebApplication.aspx.cs 源文件

图 9　解决方案资源管理器

图 10　通过转换后的测试

同时通过查看 TestWebsite.aspx.cs 文件的源代码，也会发现代码中并没有自动加入“namespace WebApplication”命名空间，但不影响运行，为了一致也可以自己手工加入命名空间，如图 11 所示。

图 11　TestWebsite.aspx.cs 源文件

## 4　结语

Website 与 WebApplication 之间的区别及相互转换就介绍到此，希望对大家有所帮助和启发，有不足之处敬请指正。

（收稿日期：2013-01-07）

# JPG 与 BMP 互转程序开发

江 洪

**摘 要**：结合 JPG 与 BMP 文件的结构，使用 VC6.0 开发了一个简单的程序。该程序支持 24 位色 JPG 与 BMP 文件之间的相互转换。通过互转程序可以了解 JPG 与 BMP 文件的结构，以及相应的压缩和解压方法，对更高层次的程序开发也有一定的参考价值。

**关键词**：JPG 文件；BMP 文件；转换

## 1 引言

JPG 文件是一种压缩图形文件格式。这种文件使用有损压缩算法对图像数据进行压缩存储，也就是说压缩后数据会有一定损失。但 JPG 压缩效率较高，而且人眼很难觉察压缩前后图片的差别，因此该文件得到了广泛的应用。

BMP 文件是一种基本的图形文件格式。图像数据一般不进行压缩，直接用 RGB 颜色分量表示每个像素。

下面开发一个实现 24 位色 JPG 与 BMP 文件相互转换的程序。

## 2 JPG 文件结构

JPG 文件由一些段组成。一般来说，JPG 文件结构如表 1 所示。

**表 1 JPG 文件结构**

| 名称 | 字段 | 长度 | 说明 |
|---|---|---|---|
| SOI<br>图像开始，标记码为 0xffd8 | | | |
| APP0<br>应用程序保留，标记码为 0xffe0 | length | 2 字节 | 段长度，不包含标记码 |
| | id | 5 字节 | 固定值 0x4a 0x46 0x49 0x46 0x00 |
| | manver | 1 字节 | 主版本，值为 1 |
| | subver | 1 字节 | 次版本，值为 1 |
| | density | 1 字节 | 0-无单位 1-点数/英寸 2-点数/厘米 |
| | xdensity | 2 字节 | X 像素密度 |
| | ydensity | 2 字节 | Y 像素密度 |
| | xthumbnail | 1 字节 | X 像素缩略图，值设 0 |
| | ythumbnail | 1 字节 | Y 像素缩略图，值设 0 |
| DQT<br>定义量化表，标记码为 0xffdb | length | 2 字节 | 段长度，不包含标记码 |
| | qtinfo | 1 字节 | 量化表信息<br>0-3 位：QT 号<br>4-7 位：QT 精度（0 8bit，1 字节；否则 16bit，2 字节） |
| | qt | 64 字节 | 64 个 8bit 精度的量化表。一般有 2 套量化表，亮度 1 套，色度 1 套 |
| SOF0<br>帧开始，标记码为 0xffc0 | length | 2 字节 | 段长度，不包含标记码，其值=8+组件数量 *3 |
| | accuracy | 1 字节 | 样本精度，设 8 |
| | height | 2 字节 | 高度，以像素为单位 |
| | weight | 2 字节 | 宽度，以像素为单位 |
| | unitcount | 1 字节 | 组件数量 1 灰度图 3 YCbCr/YIQ 彩色图 4 CMYK 彩色图 |
| | unitid | 1 字节 | 组件 ID 1=Y 2=Cb 3=Cr 4=I 5=Q |
| | samplecoff | 1 字节 | 采样系数 0-3 位：垂直采样系数<br>4-7 位：水平采样系数 |
| | qtnumber | 1 字节 | 量化表号 |
| DHT<br>定义哈夫曼表，标记码为 0xffc4 | length | 2 字节 | 段长度，不包含标记码，其值=19+n（当只有一个 HT 表时） |
| | htinfo | 1 字节 | 哈夫曼表信息<br>0-3 位：HT 号<br>4 位：HT 类型，0=DC 表，1=AC 表<br>5-7 位：必须=0 |
| | htbit | 16 字节 | HT 位表，每个数字表示相应长度哈夫曼码个数 |
| | htval | n 字节 | HT 值表<br>n=HT 位表中每个数字之和。一般有 4 套哈夫曼表，亮度的直流和交流、色度的直流和交流。<br>直流哈夫曼表中权值为后面数据位数<br>交流哈夫曼表中权值高 4 位为前面 0 的个数，低 4 位为后面数据位数 |
| SOS<br>扫描开始，标记码为 0xffda | length | 2 字节 | 段长度，不包含标记码，其值=6+2*扫描行内组件数量 |
| | unitcount | 1 字节 | 组件数量 |
| | unitid | 1 字节 | 组件 ID 1=Y 2=Cb 3=Cr 4=I 5=Q |
| | htnumber | 1 字节 | 哈夫曼表号<br>0-3 位：AC 表号 4-7 位：DC 表号 |
| | reserved | 3 字节 | 固定值 0x00 0x3f 0x00 |
| EOI<br>图像结束，标记码为 0xffd9 | | | |

JPG 文件中数据为 Motolola 格式，即高字节在前。SOS 段之后是图像数据。数据以最小编码单元（MCU）为单位进

行存放。数据末尾未用比特用 1 进行填充。因为 0xff 在 JPG 文件中用来识别段，所以如果图像数据中某个字节恰巧为 0xff，则要增加一个字节 0x00，以表示这个 0xff 是图像数据的一部分。

## 3 BMP 文件结构

BMP 文件结构如表 2 所示。

表 2 BMP 文件结构

| | 字段 | 大小 | 说明 |
|---|---|---|---|
| 文件头 | id | 2 字节 | BMP 文件标识，值为 0x42 0x4d |
| | filesize | 4 字节 | 整个 BMP 文件的字节数 |
| | reserved | 4 字节 | 保留，值为 0 |
| | dataoffset | 4 字节 | 表示位图数据距离文件头的字节数 |
| 信息头 | headsize | 4 字节 | 信息头长度，值设为 40 |
| | width | 4 字节 | 位图宽度，以像素为单位 |
| | height | 4 字节 | 位图高度，以像素为单位 |
| | planes | 2 字节 | 位平面数，值设为 1 |
| | bits | 2 字节 | 每个像素占用的位数 |
| | compress | 4 字节 | 压缩方式 0-不压缩 |
| | datasize | 4 字节 | 位图数据字节数 |
| | hresolution | 4 字节 | 水平分辨率 |
| | vresolution | 4 字节 | 垂直分辨率 |
| | colorused | 4 字节 | 使用的颜色数 |
| | colorimportant | 4 字节 | 重要的颜色数 |
| 图像数据 | data | 若干字节 | 每行数据字节数应为 4 的整数倍，如果不是 4 的倍数则用 0 进行填充，数据顺序是 BGR |

## 4 JPG 相关知识

（1）颜色模式：JPG 文件和 BMP 文件不同，使用 YCbCr 颜色，其中 Y 为亮度，Cb 为蓝色色差，Cr 为红色色差。可以使用公式将两种颜色进行转换。

BGR 转换 YCbCr 公式为：

```
y=((306*(r)+601*(g)+117*(b))>>10);
cb=((((b)-(y))*578+(128<<10))>>10);
cr=((((r)-(y))*730+(128<<10))>>10);
```

YCbCr 转换 BGR 公式为：

```
r=y+((1402*(cr-128))>>10);
g=y-((344*(cb-128))>>10)-((714*(cr-128))>>10);
b=y+((1772*(cb-128))>>10);
```

该数值还要进行范围判断，如值小于 0 则截断为 0，如值大于 255 则截断为 255。

编码时 y、cb、cr 值还要减去 128，解码时 y、cb、cr 值要加上128。

（2）最小编码单元 MCU：JPG 文件中数据是以最小编码单元为单位存放的。图像数据要划分为若干个 8*8 的矩阵。JPG 文件中，亮度采样系数为 1，即逐点扫描，而色度采样系数为 2，即隔点扫描。因此以 16*16 为一个最小编码单元。该最小编码单元中，有 4 个亮度矩阵，1 个 Cb 矩阵，1 个 Cr 矩阵。整个 BMP 文件按从左至右，从上至下地顺序划分为若干个最小编码单元。如果图像宽和高不是 16 的倍数，则需要进行扩充，以便编码。解码时，扩充的数据将被忽略。

（3）离散余弦转换：对于每个 8*8 的矩阵，要进行离散余弦转换。这是一种数学变换，分为正向 (FDCT) 和反向 (IDCT) 两种。编码时进行正向离散余弦转换，解码时进行反向离散余弦转换。通过这个转换，使得像素的变化规律呈现出来，以便编码和解码。

（4）量化：经过离散余弦转换后的矩阵要进行量化。量化即矩阵中每个数分别除以量化表中的相应值。同理，反量化即矩阵中每个数分别乘以量化表中相应值。量化是主要的有损变换的过程。

（5）矩阵系数：每个 8*8 矩阵共有 64 个系数。这 64 个系数中，第一个为直流系数 DC，其余 63 个为交流系数 AC。JPG 使用不同的方法分别为每个矩阵的 DC 和 AC 编码。

（6）ZIGZAG 扫描：对于矩阵中数据，要按一定顺序进行读取，同理，反 ZIGZAG 扫描按和 ZIGZAG 相反的顺序进行读取。

ZIGZAG 扫描顺序是：

```
0,1,8,16,9,2,3,10,17,24,32,25,18,11,4,5,
12,19,26,33,40,48,41,34,27,20,13,6,7,14,21,28,
35,42,49,56,57,50,43,36,29,22,15,23,30,37,44,51,
58,59,52,45,38,31,39,46,53,60,61,54,47,55,62,63
```

反 ZIGZAG 扫描顺序是：

```
0,1,5,6,14,15,27,28,2,4,7,13,16,26,29,42,
3,8,12,17,25,30,41,43,9,11,18,24,31,40,44,53,
10,19,23,32,39,45,52,54,20,22,33,38,46,51,55,60,
21,34,37,47,50,56,59,61,35,36,48,49,57,58,62,63
```

（7）DC 系数的编码：因为相邻两个编码单元 DC 有相差不大的特点，因此对 DC 实行差分编码，除第一个编码单元存储 DC 原值外，其他编码单元均存储当前单元和前一单元的 DC 的差值。编码时使用直流哈夫曼表。

（8）AC 系数的编码：AC 系数可能会存在许多连续的 0，根据这个特点，使用行程长度编码 RLE 进行压缩，然后使用交流哈夫曼表进行编码。遇到块结束 EOB 或已读取了 63 个 AC 则当前矩阵解码完毕。

（9）数值编码：JPG 将数值按范围分成 16 组，按照表 3 进行编码。

表 3 JPG 分组数值编码

| 实际数值 | 编码长度 | 编码 |
|---|---|---|
| 0 | 0 | - |
| -1,1 | 1 | 0,1 |
| -3,-2,2,3 | 2 | 00,01,10,11 |
| -7,-6,-5,-4,4,5,6,7 | 3 | 000,001,010,011,100,101,110,111 |
| -15,.....,-8,8,.....,15 | 4 | 0000,.....,0111,1000,.....,1111 |
| -31,.....,-16,16,.....,31 | 5 | 00000,.....,01111,10000,.....,11111 |
| -63,.....,-32,32,.....,63 | 6 | ..... |
| -127,.....,-64,64,.....,127 | 7 | ..... |
| -255,.....,-128,128,.....,255 | 8 | ..... |
| -511,.....,-256,256,.....,511 | 9 | ..... |
| -1023,.....,-512,512,.....,1023 | 10 | ..... |
| -2047,.....,-1024,1024,.....,2047 | 11 | ..... |
| -4095,.....,-2048,2048,.....,4095 | 12 | ..... |
| -8191,.....,-4096,4096,.....,8191 | 13 | ..... |
| -16383,.....,-8192,8192,.....,16383 | 14 | ..... |
| -32767,.....,-16384,16384,.....,32767 | 15 | ..... |

(10) 哈夫曼表：每个哈夫曼表，可以分为位表和值表两部分。位表 16 个字节长，每个数值表示相应长度的编码个数。比如一个值表为 0,1,5,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0 则表示长度 1 的编码没有，长度 2 的编码有 1 个，长度 3 的编码有 5 个，长度 4，5，6，7，8，9 的编码各有 1 个，没有长度 10 以上的编码。哈夫曼编码从 0 开始，相同位数的哈夫曼编码连续增加，遇到升位，最后一个低一位编码加 1 后左移一位作为高一位的第一个编码。

由上面的位表，可以确定哈夫曼码如表 4 所示。

表 4 对应分组数值编码的哈夫曼编码

| 长度 | 编码 |
|---|---|
| 2 | 00 |
| 3 | 010 |
| 3 | 011 |
| 3 | 100 |
| 3 | 101 |
| 3 | 110 |
| 4 | 1110 |
| 5 | 11110 |
| 6 | 111110 |
| 7 | 1111110 |
| 8 | 11111110 |
| 9 | 111111110 |

值表的长度为位表中所有数值之和。比如对于上面的位表，值表长度就应为 12。对于 DC 的值表，其数值表示后面数值的位数。比如 0x01 就表示后面数值为 1 位。对于 AC 的值表，其数值高 4 位表示前面 0 的个数，低 4 位表示后面数值的位数。比如 0x21 就表示前面有 2 个 0，后面数值为 1 位。值 0x00 为块结束 EOB，表示后面全是 0 了。值 0xf0 表示 16 个 0。

## 5 JPG 编码流程

(1) 划分 MCU：将整个 BMP 图像数据按从左到右，从上到下顺序划分为若干个 16*16 的方块。每行数据按照 BMP 文件中原始顺序存放。

(2) 颜色转换：BGR 颜色转换为 YCbCr。

(3) 分别对每个 MCU 编码：每个 MCU 有 4 个亮度矩阵，1 个 Cb 矩阵，1 个 Cr 矩阵。每个矩阵经过正向 DCT 转换、ZIGZAG 扫描、量化、DC 哈夫曼编码、AC 哈夫曼编码。

## 6 JPG 解码流程

(1) 以 MCU 为单位进行解码：使用哈夫曼表先解出 4 个亮度矩阵，再解出 1 个 Cb 矩阵，再解出 1 个 Cr 矩阵。

(2) 反量化、反 ZIGZAG、反向 DCT 转换：得到 1 个 DC 系数和 63 个 AC 系数。

(3) YCbCr 颜色转换 BGR：最后得到 1 个 16*16 的 BGR 图像数据。

## 7 关键代码

限于篇幅，只列出部分关键代码，其他代码可参看源程序。

MCU 编码代码如下：

```
void mcuencode(unsigned char *buf,unsigned char *qtable1,
unsigned char *qtable2,
                unsigned short huffmancount1,
                 unsigned  short  *huffmanlen1,unsigned
short *huffmancode1,
                unsigned short *huffmanvalue1,
                unsigned short huffmancount2,
                 unsigned  short  *huffmanlen2,unsigned
short *huffmancode2,
                unsigned short *huffmanvalue2,
                unsigned short huffmancount3,
                 unsigned  short  *huffmanlen3,unsigned
short *huffmancode3,
                unsigned short *huffmanvalue3,
                unsigned short huffmancount4,
                 unsigned  short  *huffmanlen4,unsigned
short *huffmancode4,
                unsigned short *huffmanvalue4,
                 unsigned long *totalbit,int *dcy,int *dccb,
```

```
int *dccr,
            unsigned char *inbuf)
{
  int buf1[1000],buf2[1000],block1[64],block2[64],block3[64];
  int y,cb,cr,b,g,r,i,j,k,l,m,n;

  for(i=1;i<=4;i++)  //4 个亮度单元编码
  {
if(i==1){m=0;n=0;}if(i==2){m=0;n=24;}
   if(i==3){m=384;n=0;}if(i==4){m=384;n=24;}
for(j=1;j<=8;j++)
   {
     l=0;
  for(k=1;k<=24;k+=3)
     {
       buf1[(j-1)*24+l]=(int)inbuf[m+(j-1)*48+n+k-1];
       buf1[(j-1)*24+l+1]=(int)inbuf[m+(j-1)*48+n+k];
       buf1[(j-1)*24+l+2]=(int)inbuf[m+(j-1)*48+n+k+1];
       l=l+3;
     }
   }
   for(j=1;j<=192;j+=3)  //bgr 颜色转换 ycbcr
   {
     b=buf1[j-1];g=buf1[j];r=buf1[j+1];
     y=((306*(r)+601*(g)+117*(b))>>10);
     cb=((((b)-(y))*578+(128<<10))>>10);
     cr=((((r)-(y))*730+(128<<10))>>10);
     if(y<0) y=0;if(y>255) y=255;
     if(cb<0) cb=0;if(cb>255) cb=255;
     if(cr<0) cr=0;if(cr>255) cr=255;
  buf2[j-1]=y-128;buf2[j]=cb-128;buf2[j+1]=cr-128;
   }
   k=0;l=0;
for(j=1;j<=64;j++){block1[k]=buf2[l];k++;l=l+3;}
   fdct(block1);zigzag(block1);quantity(block1,qtable1);
   *dcy=block1[0]-lastdcy;
   dcencode(buf,huffmancount1,
              huffmanlen1,huffmancode1,huffmanvalue1,
                *dcy,totalbit);
   lastdcy=block1[0];
   acencode(buf,huffmancount2,
              huffmanlen2,huffmancode2,huffmanvalue2,
              block1+1,totalbit);
}
j=0;k=0;
while(1)
{
  buf1[k]=inbuf[j];buf1[k+1]=inbuf[j+1];buf1[k+2]=inbuf[j+2];
     k=k+3;
     if(k%24!  =0) j=j+6;else j=j+54;
     if(k==192) break;
}
for(j=1;j<=192;j+=3)  //bgr 颜色转换 ycbcr
{
  b=buf1[j-1];g=buf1[j];r=buf1[j+1];
  y=((306*(r)+601*(g)+117*(b))>>10);
  cb=((((b)-(y))*578+(128<<10))>>10);
  cr=((((r)-(y))*730+(128<<10))>>10);
  if(y<0) y=0;if(y>255) y=255;
  if(cb<0) cb=0;if(cb>255) cb=255;
  if(cr<0) cr=0;if(cr>255) cr=255;
  buf2[j-1]=y-128;buf2[j]=cb-128;buf2[j+1]=cr-128;
}
k=0;
for(j=1;j<=192;j+=3){block2[k]=buf2[j];k++;}
k=0;
for(j=1;j<=192;j+=3){block3[k]=buf2[j+1];k++;}
//cb 单元编码
fdct(block2);zigzag(block2);quantity(block2,qtable2);
*dccb=block2[0]-lastdccb;
dcencode(buf,huffmancount3,
           huffmanlen3,huffmancode3,huffmanvalue3,
             *dccb,totalbit);
lastdccb=block2[0];
acencode(buf,huffmancount4,
           huffmanlen4,huffmancode4,huffmanvalue4,
             block2+1,totalbit);
//cr 单元编码
fdct(block3);zigzag(block3);quantity(block3,qtable2);
*dccr=block3[0]-lastdccr;
dcencode(buf,huffmancount3,
           huffmanlen3,huffmancode3,huffmanvalue3,
             *dccr,totalbit);
lastdccr=block3[0];
acencode(buf,huffmancount4,
           huffmanlen4,huffmancode4,huffmanvalue4,
             block3+1,totalbit);
}
```

MCU 解码代码如下：

```
void mcudecode(unsigned char *buf,unsigned char *qtable1,
unsigned char *qtable2,
                  unsigned short huffmancount1,
                   unsigned  short  *huffmanlen1,unsigned
short *huffmancode1,
                  unsigned short *huffmanvalue1,
                  unsigned short huffmancount2,
                   unsigned  short  *huffmanlen2,unsigned
short *huffmancode2,
                  unsigned short *huffmanvalue2,
                  unsigned short huffmancount3,
                   unsigned  short  *huffmanlen3,unsigned
short *huffmancode3,
                  unsigned short *huffmanvalue3,
                  unsigned short huffmancount4,
```

```
                    unsigned  short  *huffmanlen4,unsigned
short *huffmancode4,
                    unsigned short *huffmanvalue4,
      unsigned long *totalbit,int *dcy,int *dccb,int *dccr,
                    unsigned char *outbuf)
{
  int i,j,k,l,block[64],data1[1000],data2[1000],data3[1000];
  int y,cb,cr,b,g,r;
  for(j=1;j<=4;j++)  //4 个亮度单元解码
  {
    dcdecode(buf,huffmancount1,huffmanlen1,huffmancode1,
          huffmanvalue1,dcy,totalbit);
       block[0]=*dcy;
    acdecode(buf,huffmancount2,huffmanlen2,huffmancode2,
                huffmanvalue2,block+1,totalbit);
    antiquantity(block,qtable1);antizigzag(block);idct(block);
    if(j==1) l=0;if(j==2) l=8;if(j==3) l=128;if(j==4) l=136;
    for(k=1;k<=8;k++)
for(i=1;i<=8;i++) data1[l+16*(k-1)+i-1]=block[i-1+8*(k-1)];
  }
  //cb 单元解码
  dcdecode(buf,huffmancount3,huffmanlen3,huffmancode3,
          huffmanvalue3,dccb,totalbit);
  block[0]=*dccb;
  acdecode(buf,huffmancount4,huffmanlen4,huffmancode4,
          huffmanvalue4,block+1,totalbit);
antiquantity(block,qtable2);antizigzag(block);idct(block);
j=0;k=0;
for(i=1;i<=64;i++)
{
  data2[k+j]=block[i-1];data2[k+j+1]=block[i-1];
     data2[k+j+16]=block[i-1];data2[k+j+17]=block[i-1];
     j=j+2;if(j%16==0) k=k+16;
}
//cr 单元解码
dcdecode(buf,huffmancount3,huffmanlen3,huffmancode3,
        huffmanvalue3,dccr,totalbit);
block[0]=*dccr;
acdecode(buf,huffmancount4,huffmanlen4,huffmancode4,
          huffmanvalue4,block+1,totalbit);
antiquantity(block,qtable2);antizigzag(block);idct(block);
j=0;k=0;
for(i=1;i<=64;i++)
{
  data3[k+j]=block[i-1];data3[k+j+1]=block[i-1];
     data3[k+j+16]=block[i-1];data3[k+j+17]=block[i-1];
     j=j+2;if(j%16==0) k=k+16;
  }
  j=0;
  for(i=1;i<=256;i++)  //ycbcr 颜色转换 bgr
  {
y=data1[i-1]+128;cb=data2[i-1]+128;cr=data3[i-1]+128;
     if(y<0) y=0;if(y>255) y=255;
     if(cb<0) cb=0;if(cb>255) cb=255;
     if(cr<0) cr=0;if(cr>255) cr=255;
     r=y+((1402*(cr-128))>>10);
     g=y-((344*(cb-128))>>10)-((714*(cr-128))>>10);
     b=y+((1772*(cb-128))>>10);
     if(r<0) r=0;if(r>255) r=255;
   if(g<0) g=0;if(g>255) g=255;
   if(b<0) b=0;if(b>255) b=255;
     outbuf[j]=(unsigned char)b;
   outbuf[j+1]=(unsigned char)g;
   outbuf[j+2]=(unsigned char)r;
     j=j+3;
  }
}
```

## 8 结语

JPG 与 BMP 互转程序实现了既定的功能。希望本例程对想了解相关内容的读者有所帮助。

（收稿日期：2013-05-03）

（上接第 51 页）

```
    sum(iif((ywzj+ywfj+sxzj+sxfj+yy)>=320,1,0)) as 理应录取
人数;
    sum(iif(b2=2,1,0)) as 理应达 2B;
    from  lsx.dbf;
    where xkmc like '%物%' and kslb= '1' ;
    group by xqmc;
    order by xqdm ASC;
    INTO TABLE  各县理应模拟录取上线人数.dbf
    copy to  各县理应模拟录取上线人数.xls  type  xl5
    (10)全市四星学校总分上线人数：
    Select  xxmc  as 学校名称 ,count(*) as  理应参考人数,;
    sum(iif((ywzj+ywfj+sxzj+sxfj+yy)>=317,1,0)) as i;
    from  zk.dbf;
    where xkmc like '%物%' and kslb= '1'  and  xxxj="4";
    group by xxmc;
    order by xxmc ASC;
    INTO TABLE 85 四星理应模拟录取上线人数.dbf
    copy to 85 四星理应模拟录取上线人数.xls  type  xl5
```

## 5 结语

经过实践应用，本系统具有一定的可操作性，将原本繁琐复杂的成绩调研统计工作变得高效、快捷、方便，给出了切实有效的解决方案。

（收稿日期：2013-04-27）

# 利用 AS 3.0 实现 Flash 动画

谢建华

**摘　要**：动画是 Flash 最基本的表现形式，利用 AS 3.0 程序语言，通过不同方式实现相同的动画效果，并分析了各种方法的动画形成原理，剖析其优缺点，为 Flash 动画编程者提供了更多的编程思路。

**关键词**：AS 3.0 语言；动画；面向对象编程

## 1　AS 3.0 简介

Flash 程序语言 ActionScript 3.0（简称 AS 3.0）是基于 ECMAScript（ECMAScript 是用于脚本编写的国际标准编程语言）开发的，是一种强大的面向对象编程语言，它引入了一种新的高度优化的 ActionScript 虚拟机，称为 AVM2。与 AVM1 相比，AVM2 的性能有了显著的提高。这使 AS 3.0 代码的执行速度几乎比以前的 AS 代码快了 10 倍。

ActionScript 的老版本（AS 1.0 和 2.0）提供了创建效果丰富的 Web 应用程序所需的功能和灵活性。AS 3.0 在为基于 Web 的应用程序提供了更多的可能性。它进一步增强了这种语言，提供了出色的性能，简化了开发的过程，因此更适合高度复杂的 Web 应用程序和大数据集。AS 3.0 可以为以 Flash Player 为目标的内容和应用程序提供高性能和开发效率。

## 2　程序实现

这里利用 AS 3.0 的不同编程方式来实现一个类似的动画效果，即将一个实例名为 mBall 的小球，实现在当球运动到舞台左右边界时发生循环的反弹碰撞。并分析各种不同编程方式实现的机制，同时也可以了解到不同方式的优缺点，丰富 Flash 动画程序人员的编程思路和技巧。

### 2.1　利用 EnterFrame 事件

每当 Flash 运行器执行一次预定屏幕更新检查时，它都会调度 Event.ENTER_FRAME 事件。注册以接收 Event.ENTER_FRAME 事件的函数都被反复执行，在当前 Flash 运行器帧频决定的频率下，由任何 Event.ENTER_FRAME 事件监听器做出可见变化在它退出之间被描述：

```
addEventListener(Event.ENTER_FRAME,onEnterFrameHd);
var flag:Boolean;
function onEnterFrameHd(e:Event):void{
        //判断如果球碰到舞台的最右边或最左边,都需进行转向
//运动
      if(mBall.x>stage.stageWidth-mBall.width || mBall.x<0){
            flag = ! flag;}
      if(! flag)
            mBall.x +=5; //向右运动
      else
            mBall.x -=3;    //向左运动
}
```

Event.ENTER_FRAME 是一个跟帧频有关的侦听事件，只要程序被编译执行，它将在程序运行中一直侦听。它是把传统的 Flash 帧的“渲染帧→显示帧→渲染 next 帧→显示 next 帧”的这个流程给简化了，变成了只需一帧，通过 Event.ENTER_FRAME 就可以达到上面的效果，内存的开销大大地节省了。

### 2.2　利用 setInterval () 函数

setInterval () 函数的作用是在播放动画的时，每隔一定时间就调用函数、方法或对象。可以使用本动作更新来自数据库的变量或更新时间显示。setInterval () 函数的语法格式如下：

```
setInterval(function,interval[,arg1,arg2,.....argn])
setInterval(object,methodName,interval[,arg1,arg2,.....argn])
```

第一种格式是标准动作面板中 setInterval 函数的默认语法，第二种格式是在专家模式动作中使用的方法。

其中的参数 function 是一个函数名或者一个对匿名函数的引用。object 参数指定从 Object 对象派生的对象。methodName 制定 object 参数中要调用的方法。interval 制定对 function 或 methodName 调用两次之间的时间，单位是毫秒。后面的 arg1 等是可选的参数，用于制定传递给 function 或是 methodName 的参数：

```
var flag:Boolean,;
//每 10ms 循环执行 peng 函数,实现小球碰撞效果
setInterval(peng, 10);
function peng(){
 //判断如果球碰到舞台的最右边或最左边,都需进行转向运动
      if (mBall.x>stage.stageWidth-mBall.width || mBall.x<0)
flag=! flag;
      if(! flag)
            mBall.x +=5; //向右运动
      else
```

```
        mBall.x -=3;    //向左运动
}
```

函数 setInterval () 可以用于按照指定时间间隔进行反复不断调用某个函数，产生一种循环效果。该例中将每隔 10ms 不断地执行 peng () 函数。setInterval 它设置的时间间隔小于动画帧速（如每秒 10 帧，相当于 100 毫秒），则按照尽可能接近 interval 的时间间隔调用函数。而且必须使用 updateAfterEvent 动作来确保以足够的频率刷新屏幕。如果 interval 大于动画帧速，则只用在每次播放头进入某一帧是才调用，以减小每次刷新屏幕的影响。

## 2.3 利用 Timer 类创建动画

Timer 类用来实现按一定时间间隔来重复触发计时器，通过 new 关键字来创建新的 Timer 对象，其语法格式如下：

var timer:Timer = new Timer(delay:Number, repeatCount: int= 0);

其中，参数 delay：是读写属性，表示间隔调用的时间；repeatCount，表示当前调用的次数，0 表示调用无限次数。

使用 start () 方法来启动计时器。为 timer 事件添加事件侦听器，以便将代码设置为按计时器间隔运行；使用 stop () 用于停止调用；使用 reset () 用于重置调用。Timer 类有两个事件，当开始调用时会发生 timer 事件，调用结束时会发生 timerComplete 事件。这两个事件都是 TimerEvent 类的属性，事件名分别为 Timer.TIMER 和 Timer.TIMER_COMPLETE：

```
var flag:Boolean;
var t:Timer = new Timer(100,0);//创建 t 对象,并设置间隔为
//100ms,无限执行
t.addEventListener(TimerEvent.TIMER,onTimer);
t.start();
function onTimer(e:TimerEvent):void{
 //判断如果球碰到舞台的最右边或最左边,都需进行转向运动
      if(mBall.x>stage.stageWidth-mBall.width || mBall.x<0){
            flag = ! flag;}
      if(! flag)
            mBall.x +=5;
      else
            mBall.x -=3;
      }
```

上述代码看似 Timer 类可以提供一个完全随意的方式来在一个指定的时间间隔执行一个函数，但它依然依赖于 Flash 运行器的帧频。对于每次预定屏幕更新检查一个 TimerEvent.TIMER 事件最多可产生 10 次（10 倍于帧频）。例如，给定每秒 1 帧的帧频，一个 TimerEvent.TIMER 事件最多可以 100 毫秒执行一次，甚至在一个更小的 delay 值指定给一个 Timer 对象。

## 2.4 利用 Tween 类创建动画

Tween 类可以实现通过指定目标影片剪辑的属性在若干帧数或秒数中具有动画效果，从而对影片剪辑进行移动、调整大小和淡入淡出操作。

Tween 类还使您能够指定各种缓动方法。“缓动”是动画运行期间的渐进加速和渐进减速效果，有助于使动画显得更逼真。fl.transitions.easing 包提供了很多缓动方法（包括这种加速和减速的等式），它们会相应地更改缓动动画。

若要使用 Tween 类的这些方法和属性，请将 new 运算符用于构造函数来创建该类的一个实例，并指定一个缓动方法作为参数，其语法格式：

var myTween:Tween = new Tween (obj:Object, prop:String, func:Function, begin:Number, finish:Number, duration: Number, useSeconds:Boolean = false);

这些参数依次为：

obj：要制作动画的对象

prop：要改变对象的属性，注意这个值为字符串

func：要用何种方式去用程序补间上诉属性的动画

begin：对象属性的初始值（程序开始补间的初始值）

finish：对象属性的终端值（程序要补间到的最终值）

duration：这段补间动画持续的时间

useSeconds：设定动画持续的时间是按帧计算（useSeconds = false），还是按秒计算（useSeconds = true），默认值是使用帧数计算。

```
import fl.transitions.Tween;
import fl.transitions.easing.*;
import fl.transitions.TweenEvent;
//实现小球 1 秒时间间隔从舞台左边匀速运动到最右边
var myTween:Tween = new Tween(mBall,"x",None.easeIn, 0,
stage.stageWidth-mBall.width,1,true);
//侦听小球运动到终点时,调用 yoyo()方法
                myTween.addEventListener (TweenEvent.
MOTION_FINISH,onTweenHd);
    function onTweenHd(e:TweenEvent){
          myTween.yoyo();
          }
```

其中 fl.transitions.easing 包中有以下几种缓动效果类，可实现多种不同形式的动画效果。具体如表 1 所示。

表 1 常用缓动效果类

| 类 | 说明 |
| --- | --- |
| Back | Back 类可以定义三个缓动函数，以便实现具有 ActionScript 动画的运动。easeIn () 方法开始时往后运动，然后反向朝目标移动。easeOut () 方法开始运动时是朝目标移动，稍微过冲，再倒转方向回来朝着目标。easeInOut () 方法结合了上述两种方法的运动，开始运动时是向后跟踪，再倒转方向并朝目标移动，稍微过冲目标，然后再次倒转方向，回来朝目标移动。 |
| Bounce | Bounce 类可以定义三个缓动函数，以便实现具有 ActionScript 动画的跳动，类似一个球落向地板又弹起后，几次逐渐减小的回弹运动。easeIn () 方法以较慢速度开始回弹运动，然后在执行时加快运动速度。easeOut () 方法以较快速度开始回弹运动，然后在执行时减慢运动速度。easeInOut () 方法结合了 easeIn () 和 easeOut () 方法的运动，缓慢地开始跳动，进行加速运动，再进行减速 |

| 类 | 说明 |
|---|---|
| Elastic | Elastic 类可以定义三个缓动函数，以便实现具有 ActionScript 动画的运动，其中的运动由按照指数方式衰减的正弦波来定义。easeIn () 方法以较慢速度开始运动，然后在执行时加快运动速度。easeOut () 方法以较快速度开始运动，然后在执行时减慢运动速度。easeInOut () 方法结合了 easeIn () 和 easeOut () 方法的运动，缓慢地开始运动，进行加速运动，再进行减速。 |
| None | None 类定义缓动函数，以实现 ActionScript 动画的非加速运动。easeIn () 方法和 easeOut () 方法以匀速度运动。 |
| Regular | Regular 类可以定义三个缓动函数，以便实现具有 ActionScript 动画的加速运动。easeIn () 方法以零速率开始运动，然后在执行时加快运动速度。easeInOut () 方法兼有 easeIn () 方法和 easeOut () 方法的运动，开始运动时速率为零，先对运动进行加速，再减速直到速率为零。easeOut () 方法以较快速度开始运动，然后在执行时减慢运动速度，直至速率为零。 |
| Strong | easeIn () 方法以零速率开始运动，然后在执行时加快运动速度。easeOut () 方法以较快速度开始运动，然后在执行时减慢运动速度，直至速率为零。easeInOut () 方法兼有 easeIn () 方法和 easeOut () 方法的运动，开始运动时速率为零，先对运动进行加速，再减速直到速率为零。 |

## 3 结语

通过 AS 3.0 程序语言可以实现多种形式完成相同的动画效果，但其创建动画的原理或机制都有所不同，且每种方法对系统环境或内存都有不同的制约或影响。例如，利用 Timer 类可以创建 Timer 对象以运行一次或按指定间隔重复运行，从而按计划执行代码。取决于 SWF 文件的帧频或 Flash Player 的环境（可用内存及其他因素），Flash Player 会能会按稍有偏差的间隔调度事件。如果某个 SWF 文件设置为以每秒 10 帧 [fps]（也就是 100 毫秒的间隔）的速度播放，但计时器设置为在 80 毫秒时触发事件，则 Flash Player 将按接近于 100 毫秒的间隔触发事件，大量耗费内存的脚本也可能使事件发生偏差。

利用 Tween 类则创建出多种形式的“缓动”动画效果，使得动画效果非常逼真，执行效率也比较高，但它为 AS 3.0 的外部类，需要使用 import 语句将其导入才能实现。

### 参考文献

[1] 朱治国，缪亮，等. Flash ActionScript 3.0 编程技术教程. 清华大学出版社，2008.

[2] 刘欢. Flash ActionScript 3.0 全站互动设计. 人民邮电出版社，2010.

（收稿日期：2013-05-21）

（上接第 65 页）

完全一样，则表示程序正确，去除运行答案的每行末尾的空格，回车，还发现其他字符的不匹配，立即中止该程序，修改数据库记录为 Wrong Answer，用户可以看到自己的程序的正确性。

#### 4.3.3 评判结果准确性

为了防止学生针对答案造程序的可能性，本设计通过程序产生多组随即输入数据 (*.in) 输出数据 (*.out)，对提交程序进行多次验证。

### 4.4 管理模块

管理员可以向题库中添加各种类型且符合要求的试题，也可以对它们进行修改和删除。同时,管理员也能对用户、编程竞赛和 BBS 等数据进行管理。

(1) 题目管理：①题目录入，管理员可以手工输入也可以从文件导入。②试题修改，管理员还可以对试题进行修改。③试题删除　管理员可以删除不再需要的试题。

(2) 竞赛管理：管理员可以随时抽取题目规定时间安排编程竞赛。

### 4.5 BBS 模块

程序设计方法的提高需要大家探讨，这就是这个论坛建设的必要性。BBS 设计采用树形结构，可以通过题号进行索引浏览，使问答层次更分明。

### 4.6 数据库设计与实现

根据功能需求建立 judgeonline 数据库，主要包括 user 用户、problem 题目、solution 提交答案、message 发布信息、contest 竞赛、Cont-est_problem、news 等数据库表。

User 表字段主要有:user_id、submit、language、password、nick、school 等。

problem 表字段主要有：problem_id、title、description、input、output,hint 、source、date、time_limited、acceptec 等。

solution 主要字段有:solution_id、problem_id、user_id、time、date、result、language、contest_id、code_length 等。

message 表用与 BBS 讨论模块,主要字段有:message_id、problem_id、parent_id、depth,user_id、title、content、date 等。

## 5 结语

本系统通过 PHP，MySQL，Apache 组合实现了用户注册、更新、提交、查看状态和 BBS 讨论等功能，对 C，C++程序进行了成功编译运行。在线评测系统充分利用网络的快捷、方便，为提高同学编程能力提供了理想的平台，由此评测系统将会得到广泛应用。

### 参考文献

[1] 张恩民. PHP 开发实战权威指南. 清华大学出版社，2012.

[2] POJ. 北京大学在线评测系统 [] . http:acm.pku.edu.cn/JudgeOnline/.

[3] [美] George Reese，等. MySQL 权威指南. 林琪，等，译. 中国电力出版社，2003.

（收稿日期：2013-04-19）

# AS3 中实现汉诺塔问题递归算法动画显示

程海生

**摘　要**：基于 Flash AS3 开发环境，实现了将汉诺塔问题递归算法的结果以动画的形式显示输出。

**关键词**：教育技术；Flash 软件；AS3 编程；汉诺塔；递归算法；动画

## 1　问题由来

在印度，有这么一个古老的传说：在世界中心贝拿勒斯（在印度北部）的圣庙里，一块黄铜板上插着 3 根宝石针。印度教的主神梵天在创造世界的时候，在其中一根针上从下到上地穿好了由大到小的 64 片金片（如图 1 所示），这就是所谓的汉诺塔。不论白天黑夜，总有一个僧侣在按照下面的法则移动这些金片：一次只移动一片，不管在哪根针上，小片必须在大片上面。僧侣们预言，当所有的金片都从梵天穿好的那根针上移到另外一根针上时，世界就将在一声霹雳中消灭，而梵塔、庙宇和众生也都将同归于尽。

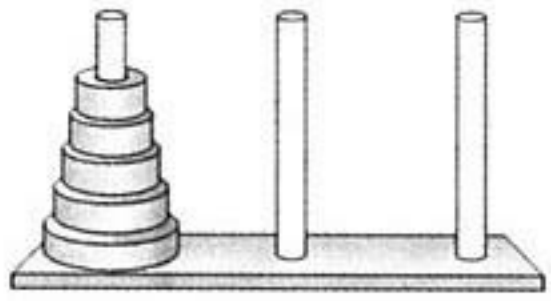


图 1　汉诺塔模型

## 2　意义

汉诺塔问题是程序设计中的经典递归问题，以动画的形式呈现汉诺塔问题对于初学者学习递归算法有一定帮助作用。

## 3　准备工作

本例在 Flash CS5.5 中完成，编写程序前首先做如下准备工作：

（1）插入一个影片剪辑元件，在其中绘制一个圆盘放在中心位置，在元件属性中选中为“ActionScript 导出”，类为 mc，如图 2 所示。

（2）在舞台上插入一个 NumericStepper 组件，在属性面板中将其命名为 panzishu，用于选择圆盘多少。因为圆盘多运行的时间会很长，因此将实例中圆盘最大数设为 9，当然如果需要体现移动的难度也可设置更大的数。

（3）在舞台上插入一个 Button 组件，在属性面板中将其命名为 kaishi，将其属性中的 label 值改为“开始演示”。


图 2　创建影片剪辑元件

（4）在舞台适当位置放置 3 个立方体并分别标注 A、B、C，代表 3 个柱子。

## 4　算法实现

在第一帧上输入如下语句：

```
import flash.events.Event;  //连同以下几句导入必要的类
import flashx.textLayout.utils.CharacterUtil;
import flash.text.TextField;
import flash.utils.Timer;
import flash.events.TimerEvent;
var i:uint = 0;  //保存圆盘数
var a = [];
a[1]=new Array();  //保存 a 柱上的圆盘 mc
a[2]=new Array();  //保存 b 柱上的圆盘 mc
a[3]=new Array();  //保存 c 柱上的圆盘 mc
var get1 = []; //保存移动的起始柱
var put1 = []; //保存移动的目标柱
var score:uint; //移动总次数
var tempMc;//临时保存 mc
panzishu.addEventListener(Event.CHANGE,changePanzishu);
function changePanzishu(e:Event)    //用于使微调组件改变圆
//盘个数
{
    if (e.target.value > i)
    {
        tempMc=new mc();
        tempMc.x = 120;
        if (i==0)
        {
```

GAME PROGRAM

```
            tempMc.y = 320;
        }
        else
        {
            tempMc.y = a[1][i - 1].y - a[1][i - 1].height
* 0.3;//根据下面圆盘位置确定上面圆盘的坐标 y
            tempMc.width = a[1][i - 1].width * 0.9;//连同
//下句使上面的圆盘是下面相临圆盘大小的 90%
            tempMc.height = a[1][i - 1].height * 0.9;
        }
        addChild(tempMc);
        a[1].push(tempMc);
        i++;
    }
    else
    {
        removeChild(a[1].pop());
        i--;
    }
}
kaishi.addEventListener(MouseEvent.CLICK,hanoi);
function hanoi(e:MouseEvent)
{
    hanoi2(i,1,2,3);
    var ti:uint;
    var x1:Number,x2:Number,y1:Number,y2:Number;
    var t:Timer = new Timer(800,score);
    t.addEventListener(TimerEvent.TIMER,show1);
    t.start();
    function show1(event:TimerEvent)  //形成整个动画
    {
        x1 = a[get1[ti]][a[get1[ti]].length - 1].x;
        y1 = a[get1[ti]][a[get1[ti]].length - 1].y;
        if (a[put1[ti]].length > 0)
        {
            y2 = a[put1[ti]][a[put1[ti]].length - 1].y - a
[put1[ti]][a[put1[ti]].length - 1].height * 0.3;
            x2 = a[put1[ti]][a[put1[ti]].length - 1].x;
        }
        else
        {//如果目的柱上没有圆盘时,下面几句给出柱上第一
//个圆盘的坐标
            if (put1[ti] == 1)
            {
                x2 = 120;
                y2 = 320;
            }
            else if (put1[ti]==2)
            {
                x2 = 320;
                y2 = 210;
            }
            else
            {
                x2 = 460;
                y2 = 330;
            }
        }
        tempMc = a[get1[ti]].pop();
        a[put1[ti]].push(tempMc);
        ti++;
        var t2:Timer = new Timer(70,8);
        t2.addEventListener(TimerEvent.TIMER,show2);
        t2.start();
        function show2(event:TimerEvent) //形成一次移动动画
        {
            tempMc.x +=  (x2 - x1) / 8;
            tempMc.y +=  (y2 - y1) / 8;
        }
    }
}
function hanoi2(n:uint,one:uint,two:uint,three:uint)
{
    if (n==1)
    {
        move1(one,three);
    }
    else
    {
        hanoi2(n-1,one,three,two);
        move1(one,three);
        hanoi2(n-1,two,one,three);
    }
}
function move1(getone:uint,putone:uint)
{
    score++;
    get1.push(getone);
    put1.push(putone);
}
stop();
```

最终效果如图 3 所示。

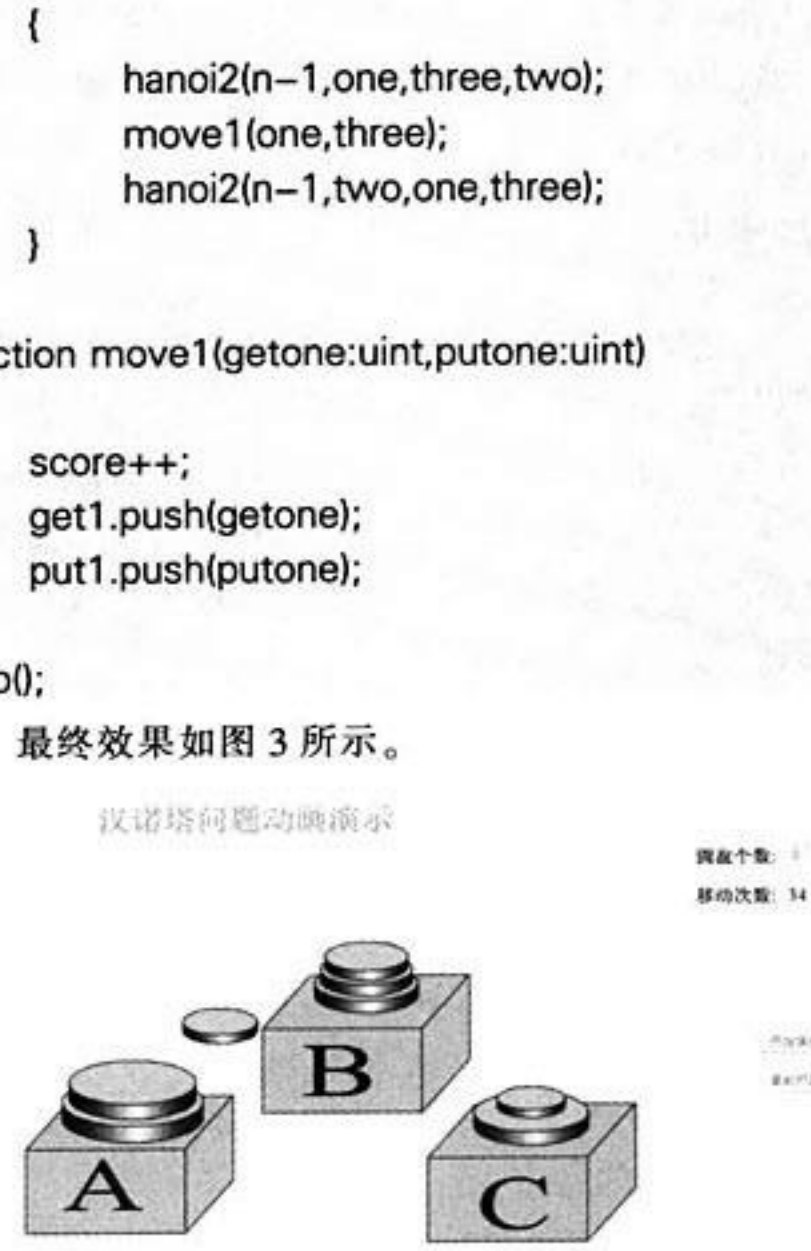



图 3　运行效果

（收稿日期：2013-04-16）

# 用 VC++ 开发 APE 文件播放和转码软件

赵常寿 张 鹏 韦宏强

**摘 要**：利用 APE 开发库提供的动态链接库 MACDLL 对 APE 格式的音频文件进行解码，实现 APE 文件的播放和转码为 WAV 文件。

**关键词**：APE 文件； WAV 文件； MACDLL 库； 解码器

## 1 引言

APE 是流行的数字音乐文件格式之一，与 MP3 这类有损压缩方式不同，APE 是一种无损压缩音频技术，也就是说从音频 CD 上读取的音频数据文件压缩成 APE 格式后，再将 APE 格式的文件还原，还原后的音频文件与压缩前的一模一样，没有任何损失。因此常用于保存高品质的音频数据，比如 CD 光盘常用 APE 格式压缩，既缩小了体积又不损失音质，方便存储和传播。但是，APE 格式需要解码才能进行声音输出，而很多软件不支持这种格式的解码，比如著名的刻录软件 NERO 不支持 APE 格式，需要安装专门的插件或者将 APE 转码为其能识别的格式才能处理。下文通过开发一个软件，实现 APE 文件的播放和转码为 WAV 文件。

## 2 APE 开发库

开发者可以利用 APE 官方网站提供的 SDK 开发包进行 APE 格式的编码和解码，该开发包的源代码是公开的，可以免费下载，下载网址为 http://www.monkeysaudio.com/index.html，目前最新版本为 v4.11。开发包中提供了两种编解码器使用方式，一种是 LIB 静态链接库，另一种是 DLL 动态链接库，这里采用动态链接库方式。

动态链接库文件是 MACDLL.DLL，提供的解码部分 API 如下：

### 2.1 c_APEDecompress_Create

函数功能是打开 APE 文件，创建解码器，函数原型如下：

```
APE_DECOMPRESS_HANDLE c_APEDecompress_Create ( const str_ansi * pFilename, int * pErrorCode = NULL) ;
```

其中，第一个参数是 APE 文件名；第二个参数是错误代码。返回值是 APE 解码器的句柄。

### 2.2 c_APEDecompress_Destroy

函数功能是关闭 APE 文件，销毁解码器，函数原型如下：

```
void c_APEDecompress_Destroy(APE_DECOMPRESS_HANDLE hAPEDecompress);
```

其中，唯一的参数是 APE 解码器的句柄。

### 2.3 c_APEDecompress_GetData

函数功能是解码音频块，函数原型如下：

```
int c_APEDecompress_GetData ( APE_DECOMPRESS_HANDLE hAPEDecompress, char * pBuffer, int nBlocks, int * pBlocksRetrieved);
```

其中，第一个参数是 APE 解码器的句柄；第二个参数是解码后的数据缓冲区；第三个参数是需要解码的数据块数；第四个参数是实际解码的数据块数。

### 2.4 c_APEDecompress_Seek

函数功能是定位音频块，函数原型如下：

```
int c_APEDecompress_Seek (APE_DECOMPRESS_HANDLE hAPEDecompress, int nBlockOffset);
```

其中，第一个参数是 APE 解码器的句柄；第二个参数是数据块的偏移量。

### 2.5 c_APEDecompress_GetInfo

函数功能是获取 APE 文件和解码器的相关信息，函数原型如下：

```
int c_APEDecompress_GetInfo(APE_DECOMPRESS_HANDLE hAPEDecompress, APE_DECOMPRESS_FIELDS Field, int nParam1 = 0, int nParam2 = 0);
```

其中，第一个参数是 APE 解码器的句柄；第二个参数是需要获取的信息的编号；第三和第四个参数是需要获取的信息的参数。

## 3 WAVE 文件格式

WAVE 文件是计算机领域最常用的数字化声音文件格式之一，它是微软专门为 Windows 系统定义的波形文件格式(Waveform Audio)，其扩展名为" *.wav"，它采用 RIFF 文件格式结构，WAVE 文件格式定义如下。

### 3.1 文件头

首先是文件头定义如表 1 所示。

表 1　WAVE 类文件结构

| 偏移地址 | 字节数 | 数据类型 | 内 容 |
|---|---|---|---|
| 00H | 4 | char | " RIFF" 标志 |
| 04H | 4 | unsigned int | 文件长度-8 |
| 08H | 4 | char | " WAVE" 标志 |
| 0CH | 4 | char | " fmt " 标志 |
| 10H | 4 | unsigned int | 编码格式长度，16 |
| 14H | 2 | unsigned short | 声音编码类型，1 |
| 16H | 2 | unsigned short | 声道数 |
| 18H | 4 | unsigned int | 采样频率 |
| 1CH | 4 | unsigned int | 平均传输速率 |
| 20H | 2 | unsigned short | 数据块对齐字节数 |
| 22H | 2 | unsigned short | 样本大小 |
| 24H | 4 | char | " data" 标志 |
| 28H | 4 | unsigned int | WAV 数据长度 |

## 3.2　PCM 波形数据

文件头后面紧接着就是声音数据，WAV 数据的每个样本值包含在一个整数中，其长度为容纳指定样本大小所需的最小字节数，8 位和 16 位的 PCM 波形数据格式如表 2 所示。

表 2　8 位和 16 位的 PCM 波形数据格式

| 样本大小 | 数据格式 | 最小值 | 最大值 |
|---|---|---|---|
| 8 位 PCM | unsigned char | 0 | 255 |
| 16 位 PCM | short | -32768 | 32767 |

PCM 波形数据的存储方式如下：

8 位单声道：声道 0　声道 0

8 位双声道：声道 0(左)　声道 1(右)　声道 0(左)　声道 1(右)

16 位单声道：声道 0 低字节　声道 0 高字节　声道 0 低字节　声道 0 高字节

16 位双声道：声道 0(左)低字节 声道 0(左)高字节 声道 1(右)低字节 声道 1(右)高字节

# 4　关键代码

程序使用 Visual C++ 6.0 开发，主要功能有两个：一是播放 APE 文件；二是转码 APE 文件为 WAV 文件。程序界面如图 1 所示。

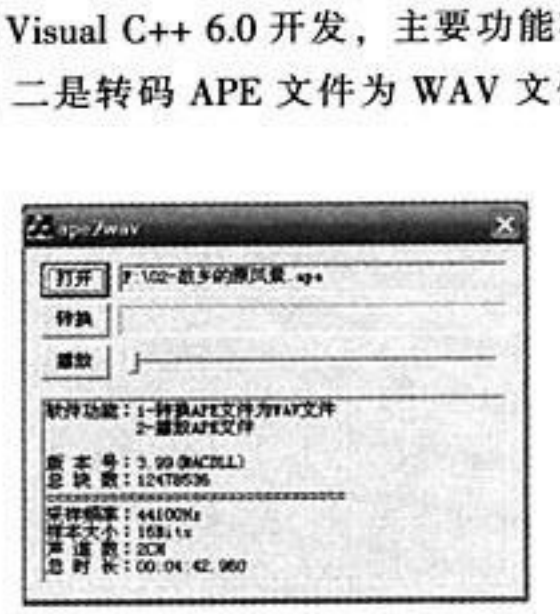



图 1　程序运行界面

## 4.1　打开 APE 文件

```
//打开 Ape 文件
void CApe2wavDlg::OnButton1()
{
	int ret;
	int hh,mm,ss,mms;
	CString str1,str2;
	const char szFilters[]="Ape Files (*.ape)|*.ape||";
		CFileDialog dlg (TRUE, "*.*", NULL,
OFN_FILEMUSTEXIST| OFN_HIDEREADONLY, szFilters);
	if(dlg.DoModal() == IDOK) //打开文件对话框
	{
		fn1=dlg.GetPathName();  //APE 文件名
		fn2=fn1+".wav";      //WAV 文件名
		m_EDIT1.SetWindowText(fn1);
		if(pAPE)
		{
		c_APEDecompress_Destroy(pAPE); //销毁解码器
		  pAPE=NULL;
		}
		pAPE =c_APEDecompress_Create ((LPCTSTR)fn1,
&ret);//创建解码器
		if (pAPE==NULL)
		{
		  AfxMessageBox("不是 APE 文件");
		  return;
		}
str2.Format (" 软件功能：1-转换 APE 文件为 WAV 文件 \r\n
2-播放 APE 文件 \r\n\r\n");
		str1=str2;
		str2.Format (" 版 本 号：% 4.2lf (MACDLL)\r\n",
c_APEDecompress_GetInfo(pAPE,APE_INFO_FILE_VERSION)
/1000.0);
		str1=str1+str2;
str2.Format (" 总 块 数：% d\r\n",c_APEDecompress_GetInfo
(pAPE,APE_INFO_TOTAL_BLOCKS));
		str1=str1+str2;
	  str1=str1+"============================\r\n";
str2.Format("采样频率：%dHz\r\n",c_APEDecompress_GetInfo
(pAPE,APE_INFO_SAMPLE_RATE));
		str1=str1+str2;
str2.Format("样本大小：%dBits\r\n",c_APEDecompress_GetInfo
(pAPE,APE_INFO_BITS_PER_SAMPLE));
		str1=str1+str2;
				str2.Format (" 声 道 数：%dCH\r\n",
c_APEDecompress_GetInfo(pAPE,APE_INFO_CHANNELS));
		str1=str1+str2;
			mms=c_APEDecompress_GetInfo(pAPE,
APE_INFO_LENGTH_MS);
		ss=(mms/1000)%60;
		mm=(mms/1000/60)%60;
		hh=mms/1000/60/60;
```

```
        mms=mms%1000;
        str2.Format (" 总 时 长 :% 02d:% 02d:% 02d.%
03d\r\n",hh,mm,ss,mms);
        str1=str1+str2;
        m_EDIT2.SetWindowText(str1);
        m_BUTTON2.EnableWindow(TRUE);
        m_BUTTON3.EnableWindow(TRUE);
        wavout.Open(); //打开音频输出设备
    }
}
```

### 4.2 转换 APE 文件

```
//转换
void CApe2wavDlg::OnButton2()
{
    if(bTransform)
        bTransform=FALSE;
    else
    {
        bTransform=TRUE;
        m_BUTTON2.SetWindowText("停止");
        m_BUTTON1.EnableWindow(FALSE);
        m_BUTTON3.EnableWindow(FALSE);
        CString str1;
        CFile wavfile;   //用于保存 WAV 文件
    WAVE_HEADER wavheader;//WAVE 文件头
            int nBlockAlign = c_APEDecompress_GetInfo
(pAPE, APE_INFO_BLOCK_ALIGN);//获取块对齐字节数
        int nTotalBlocks = c_APEDecompress_GetInfo
(pAPE, APE_DECOMPRESS_TOTAL_BLOCKS);//获取需要解
//码的总块数
        int nBlocksRetrieved = 1;
        int nTotalBlocksRetrieved = 0;
        int nRetVal=0;
        unsigned char * pBuffer = new unsigned char
[1024000 * nBlockAlign];//解码后数据缓冲区
        MSG msg;
        //打开 WAVE 文件
          wavfile.Open ((LPCTSTR)fn2,CFile::modeWrite |
CFile::modeCreate);
    //WAVE 文件头
    wavheader.cRIFFHeader[0]='R';
    wavheader.cRIFFHeader[1]='I';
    wavheader.cRIFFHeader[2]='F';
    wavheader.cRIFFHeader[3]='F';
    wavheader.nRIFFBytes=0;
    wavheader.cDataTypeID[0]='W';
    wavheader.cDataTypeID[1]='A';
    wavheader.cDataTypeID[2]='V';
    wavheader.cDataTypeID[3]='E';
    wavheader.cFormatHeader[0]='f';
    wavheader.cFormatHeader[1]='m';
    wavheader.cFormatHeader[2]='t';
    wavheader.cFormatHeader[3]=' ';
    wavheader.nFormatBytes=16;
    wavheader.nFormatTag=WAVE_FORMAT_PCM;
                wavheader.nChannels =
c_APEDecompress_GetInfo(pAPE,APE_INFO_CHANNELS);
      wavheader.nSamplesPerSec =c_APEDecompress_GetInfo
(pAPE,APE_INFO_SAMPLE_RATE);
    wavheader.nBitsPerSample =c_APEDecompress_GetInfo
(pAPE,APE_INFO_BITS_PER_SAMPLE);
                wavheader.nBlockAlign =wavheader.
nBitsPerSample*wavheader.nChannels/8;
            wavheader.nAvgBytesPerSec =wavheader.
nSamplesPerSec*wavheader.nBlockAlign;
        wavheader.cDataHeader[0]='d';
        wavheader.cDataHeader[1]='a';
        wavheader.cDataHeader[2]='t';
        wavheader.cDataHeader[3]='a';
        wavheader.nDataBytes=0;
        wavfile.SeekToBegin();
                wavfile.Write (&wavheader,sizeof
(WAVE_HEADER));//写 WAVE 文件头
        m_PROGRESS1.SetRange32(0,nTotalBlocks);
//设置进度条
        c_APEDecompress_Seek(pAPE,0);   //定位数据块
        while (nBlocksRetrieved > 0)
        {
    while(PeekMessage(&msg,0,0,0,PM_REMOVE))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
        if(bTransform==FALSE)
            break;
      nRetVal = c_APEDecompress_GetData (pAPE,
(char*)pBuffer, 1024000, &nBlocksRetrieved);  //解码数据块
        if (nRetVal ! = 0)
        {
            AfxMessageBox("转换失败");
            break;
        }
        nTotalBlocksRetrieved += nBlocksRetrieved;
        m_PROGRESS1.SetPos
(nTotalBlocksRetrieved);   //更新进度条
                wavfile.Write (pBuffer,
nBlocksRetrieved*nBlockAlign);//写 PCM 波形数据
        }
        delete []pBuffer;
                wavheader.nDataBytes =
nTotalBlocksRetrieved*nBlockAlign;
    wavheader.nRIFFBytes=wavheader.nDataBytes+36;
```

(下转第 93 页)

# 本地打印审计实现方案分析

李朝中

**摘　要：** 主要介绍微软 Windows 系统的打印组件、打印流程和打印内容相关审计方案。

**关键词：** 打印审计；打印组件；打印流程；Windows 系统

## 1　引言

在一些终端内控系统的实现中有时需要对终端用户的打印行为做审计，其中对打印内容的审计成为一个重点。下面就来分析一个打印内容审计的方案。这里所讨论的打印内容审计方案只针对微软 Windows 2000 及之后系统。微软 Windows 2000 及之后系统对打印体系结构制定一个标准流程，下面介绍打印体系结构，打印流程和两个可做打印内容审计的监视点。

## 2　Windows 打印体系

### 2.1　打印体系结构

Windows 2000 及之后的系统中打印体系结构包含了打印假脱机服务和一系列的打印驱动组件。应用程序通过调用独立于设备的 Win32 和 GDI 打印函数，可以创建打印作业并将它们发送到各种设备，如激光打印机、矢量绘图仪、光栅打印机和传真机。打印驱动需要提供用户接口组件；允许用户控制、配置打印机可选项。一个应用程序调用 Win32 GDI 函数，传递到 GDI 图形引擎，GDI 图形引擎可以自己生成假脱机结构 EMF 文件，也可以与打印驱动协作生成可被打印的图像，并递交给打印假脱机服务。如果 GDI 图形引擎生成的是假脱机结构的 EMF 文件则打印假脱机组件将会解析 EMF 文件，同时向数据流中插入页布局信息和作业控制指令。然后再由假脱机服务发送数据流到串口、并口、或者连接着目的打印机 I/O 端口的网络端口驱动。所有的假脱机组件和驱动组件被设计成可替换的，所以硬件厂商能够很容易地为新的硬件设备提供支持。

### 2.2　假脱机服务组件

假脱机服务组件包括客户端接口（Winspool.drv）、打印提供者、打印处理器、打印监视器。应用程序通过调用客户端接口来开展打印工作；客户端接口通过 RPC 方式把请求转发给假脱机服务（Spoolsv.exe）；Spoolss.dll 提供一个到打印提供者（Localspl.dll,Win32spl.dll）的路由功能；打印提供者提供一个到打印监视器的路由功能；打印监视器的一个实例对应一个物理打印机，请求最终由监视器提交给打印机。下面是 WDK 帮助文档中关于打印相关组件协作流程，如图 1 所示。

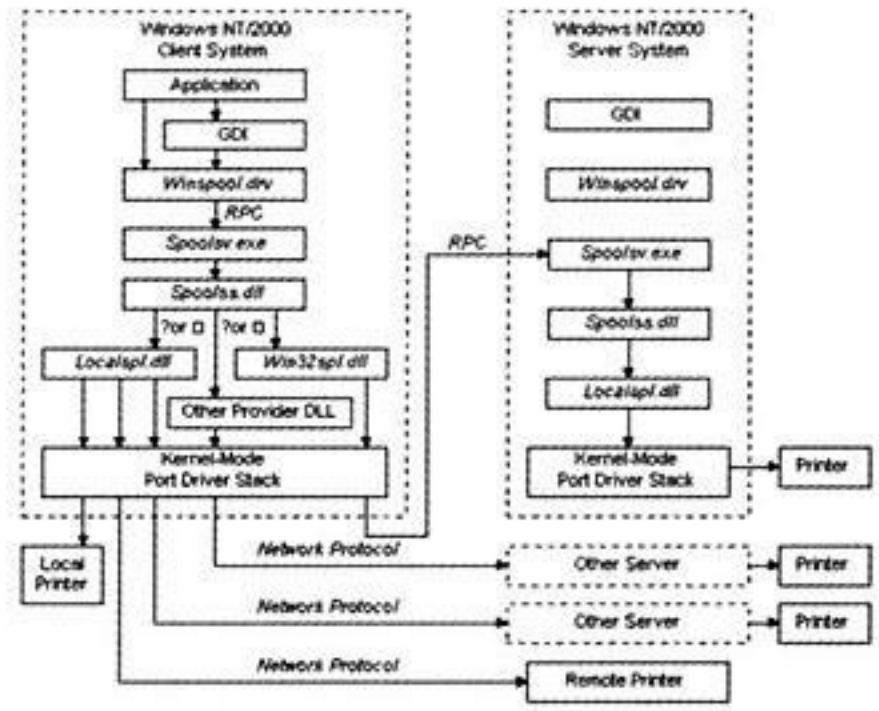



**图 1　打印相关组件协作流程**

### 2.3　打印流程实例

上面的两个小节简述了 Windows 2000 及之后系统的打印体系结构和假脱机相关的组件及协作流程。由于篇幅原因这里不能详述其全部内容，要更详细地了解打印体系结构相关的内容，可以阅读 WDK 帮助文档中关于打印驱动的相关内容。下面这个 WDK 帮助文档中的图例完整说明了一个本地打印的流程（如图 2 所示）。以下的小节会详细说明这个过程。

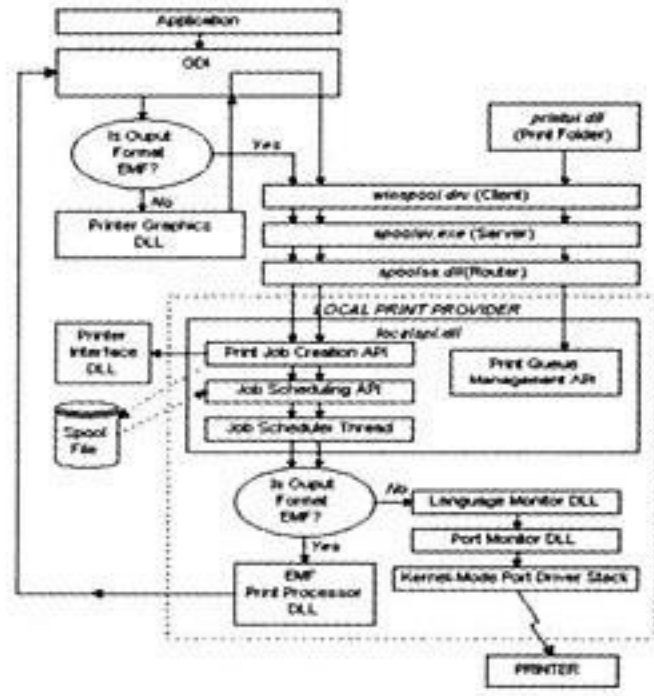



**图 2　本地打印流程**

### 2.3.1 本地打印流程

应用程序直接调用打印客户端口或通过 GDI 将请求转发到假脱机服务。假脱机服务会路由来决定交给本地打印提供都还是远程打印提供者。接下来看看本地打印提供者处理的情况。当有数据到来时假脱机服务会检查数据的类型，并根据这个类型来调用相关的打印处理器。下面是一个具体打印过程中打印处理器的函数栈：

```
ChildEBP RetAddr  Args to Child
063df19c 73d542b7 05f3e188 063df30c 00000000 winprint!
PrintDocumentOnPrintProcessor
063df1bc 6149c35c 0663737c 063df30c 00b06a4a
PrintIsolationProxy! sandbox::PrintProcessor::PrintDocThroug
hPrintProcessor+0x49 (FPO: [Non-Fpo])
063df228 6149b052 00ad6744 063df30c 00b06a4a localspl!
sandbox::PrintProcessorExecuteObserver::
PrintDocThroughPrintProcessor+0x84 (FPO: [Non-Fpo])
063df244 61499d20 00ad6744 063df30c 00b06a4a localspl!
sandbox::PrintProcessorAdapterImpl::PrintDocumentOnPrint
Processor+0x27 (FPO: [Non-Fpo])
063df270 61460400 00aa3960 063df30c 0018c938 localspl!
sandbox::PrintProcessorAdapter::
PrintDocumentOnPrintProcessor+0x77 (FPO: [Non-Fpo])
063df950 614615fe 00aa4f20 063df970 00000000 localspl!
PrintDocumentThruPrintProcessor+0x3bd
063dfbb0 76161174 00aa4f20 063dfbfc 77d1b3f5 localspl!
PortThread+0x41c (FPO: [Non-Fpo])
063dfbbc 77d1b3f5 00aa4f20 70896cd8 00000000 kernel32!
BaseThreadInitThunk+0xe (FPO: [Non-Fpo])
063dfbfc 77d1b3c8 614611e2 00aa4f20 00000000 ntdll!
_RtlUserThreadStart+0x70 (FPO: [Non-Fpo])
063dfc14 00000000 614611e2 00aa4f20 00000000 ntdll!
_RtlUserThreadStart+0x1b (FPO: [Non-Fpo])
```

在这个栈中 PrintDocumentOnPrintProcessor 是打印处理器中最重要的一个接口，它负责打印数据的格式转换，WinPrint.Dll 是 Win7 系统中微软自带的一个通用打印处理器，支持 EMF，RAW，TXT 这几种格的转换。大家注意一下 05f3e188 这个数字，这是负责本次打印的打印机的句柄。后面根据这个来跟踪这次打印的过程。

本地打印提供者导出了一个函数 SplWriterPrinter，它负责向打印机监视器传递打印机能识别的 RAW 数据流。下面对它下个断点跟踪它的调用。本地打印处理器的 SplWriter Printer 函数调用的栈信息如下：

```
ChildEBP RetAddr  Args to Child
063de2f0 00db9295 05f3cba8 01b70000 00002b64 localspl!
SplWritePrinter (FPO: [Non-Fpo])
063de308 00da8f7e 05f20620 01b70000 00002b64 spoolsv!
WritePrinter+0x2c (FPO: [Non-Fpo])
063de324 738bc27a 05f20620 01b70000 00002b64 spoolsv!
YWritePrinter+0x2b (FPO: [Non-Fpo])
063de374 738c150e 003991f4 00000000 7b03d19b
winspool! FlushBuffer+0xee (FPO: [Non-Fpo])
063de3c4 6d57cf9f 003991f4 00000000 00000002 winspool!
GetJobW+0x36 (FPO: [Non-Fpo])
WARNING: Stack unwind information not available.
Following frames may be wrong.
063de57c 6d57de7e 00d40008 063de598 00d40008
hpvpldrv09+0xcf9f
063de5a4 6d577f25 00d40008 06c30034 ffffffff hpvpldrv09+
0xde7e
063de5c0 64e5ee13 00d40008 06c30034 ff7fffff
hpvpldrv09+0x7f25
063de5d8 64e5fcc4 00d40008 06c30034 06499738 UNIDRV!
DrvEnableDriver+0x5401
063de5f8 763d1a16 00d40008 06c30034 063def08 UNIDRV!
DrvEnableDriver+0x62b2
063de630 77e43829 063dee98 063de648 00000004 GDI32!
GdiPrinterThunk+0x753 (FPO: [Non-Fpo])
063dee80 77d0642e 063dee98 0000001c 063def20 USER32!
_ClientPrinterThunk+0x28 (FPO: [Non-Fpo])
063deeb0 76406825 763f7c2c b42103eb 00000000 ntdll!
KiUserCallbackDispatcher+0x2e (FPO: [0,0,0])
063deeb4 763f7c2c b42103eb 00000000 063def08 GDI32!
NtGdiDoBanding+0xc (FPO: [4,0,0])
063deee0 763f9aed b42103eb 063def08 a3ccdf2a GDI32!
NextBand+0x1d (FPO: [Non-Fpo])
063def30 73784a81 003260f8 00000001 00a9fd78 GDI32!
GdiEndPageEMF+0x93 (FPO: [Non-Fpo])
063def74 737852bc 003260f8 b42103eb 00000000 winprint!
PrintOneSideReverseEMF+0x1be (FPO: [Non-Fpo])
063defb4 7378533c 003260f8 b42103eb 737848c3 winprint!
PrintEMFInPredeterminedOrder+0xc8 (FPO: [Non-Fpo])
063defdc 737854ba 003260f8 b42103eb 063df0a4 winprint!
PrintEMFSingleCopy+0x4d (FPO: [Non-Fpo])
063df058 737859a9 003260f8 b42103eb 063df0a4 winprint!
BPrintEMFJobNow+0x175 (FPO: [Non-Fpo])
063df18c 737834db 05f3e188 00326134 063df1bc winprint!
PrintEMFJob+0x48f (FPO: [Non-Fpo])
063df19c 73d542b7 05f3e188 063df30c 00000000 winprint!
PrintDocumentOnPrintProcessor+0x3c (FPO: [Non-Fpo])
063df1bc 6149c35c 0663737c 063df30c 00b06a4a PrintIso
lationProxy! sandbox::PrintProcessor::PrintDocThroughPrintPr
ocessor+0x49 (FPO: [Non-Fpo])
063df228 6149b052 00ad6744 063df30c 00b06a4a localspl!
sandbox::PrintProcessorExecuteObserver::
PrintDocThroughPrintProcessor+0x84 (FPO: [Non-Fpo])
063df244 61499d20 00ad6744 063df30c 00b06a4a localspl!
sandbox::PrintProcessorAdapterImpl::
PrintDocumentOnPrintProcessor+0x27 (FPO: [Non-Fpo])
063df270 61460400 00aa3960 063df30c 0018c938 localspl!
sandbox::PrintProcessorAdapter::
PrintDocumentOnPrintProcessor+0x77 (FPO: [Non-Fpo])
```

063df950 614615fe 00aa4f20 063df970 00000000 localspl!
PrintDocumentThruPrintProcessor+0x3bd (FPO: [Non-Fpo])
063dfbb0 76161174 00aa4f20 063dfbfc 77d1b3f5 localspl!
PortThread+0x41c (FPO: [Non-Fpo])
063dfbbc 77d1b3f5 00aa4f20 70896cd8 00000000 kernel32!
BaseThreadInitThunk+0xe (FPO: [Non-Fpo])
063dfbfc 77d1b3c8 614611e2 00aa4f20 00000000 ntdll!
_RtlUserThreadStart+0x70 (FPO: [Non-Fpo])
063dfc14 00000000 614611e2 00aa4f20 00000000 ntdll!
_RtlUserThreadStart+0x1b (FPO: [Non-Fpo])

根据打印机句柄知道是在打印处理器的 PrintDocumentOnPrintProcessor 函数中看到那个打印任务。而且还可以从栈信息中到这个函数 PrintDocumentOnPrintProcessor 的调用。再来看看 SplWriterPrinter 函数的原型，它是打印提供者必须提供的一个接口（关于打印处理器、打印提供者、打印监视器所在提供什么接口请参考 WDK 帮助文档）：

```
BOOL WritePrinter(HANDLE hPrinter, // 打印句柄
  LPVOID pBuf, // 写入的缓冲区
  DWORD cbBuf, // 写入的缓冲区长度
  LPDWORD pcWritten); // 返回写入的长度
```

根据上面对参数的介绍，第二个参数里存储的就是要传递给打印监视器的数据流。把第二个参数 DUMP 出来，如图 3 所示。

```
00000000h: 1B 25 2D 31 32 33 34 35 58 40 50 4A 4C 20 53 45 ; .%-12345X@PJL SE
00000010h: 54 20 53 54 52 49 4E 47 43 4F 44 45 53 45 54 3D ; T STRINGCODESET=
00000020h: 55 54 46 38 0A 40 50 4A 4C 20 4A 4F 42 20 4E 41 ; UTF8.@PJL JOB NA
00000030h: 4D 45 3D 22 69 6D 61 67 65 32 2E 6C 6F 67 20 2D ; ME="image2.log -
00000040h: 20 E8 AE B0 E4 BA 8B E6 9C AC 22 0A 40 50 4A 4C ;  璁板事鏈?.@PJL
00000050h: 20 43 4F 4D 4D 45 4E 54 3D 22 68 70 20 4C 61 73 ;  COMMENT="hp las
00000060h: 65 72 4A 65 74 20 31 31 36 30 20 28 36 31 2E 36 ; erJet 1160 (61.6
00000070h: 33 2E 34 36 31 2E 34 32 29 3B 20 57 69 6E 64 6F ; 3.461.42); Windo
00000080h: 77 73 20 37 20 55 6C 74 69 6D 61 74 65 20 36 2E ; ws 7 Ultimate 6.
00000090h: 31 2E 37 36 30 30 2E 31 3B 20 55 6E 69 64 72 76 ; 1.7600.1; Unidrv
000000a0h: 20 30 2E 33 2E 37 36 30 30 2E 31 36 33 38 35 22 ;  0.3.7600.16385"
000000b0h: 0A 40 50 4A 4C 20 43 4F 4D 4D 45 4E 54 3D 22 55 ; .@PJL COMMENT="U
000000c0h: 73 65 72 4E 61 6D 65 3A 20 57 69 6C 6C 69 6E 67 ; serName: Willing
000000d0h: 42 75 67 3B 20 41 70 70 20 46 69 6C 65 4E 61 6D ; Bug; App FilenAm
000000e0h: 65 3A 20 69 6D 61 67 65 32 2E 6C 6F 67 20 2D 20 ; e: image2.log -
000000f0h: E8 AE B0 E4 BA 8B E6 9C AC 3B 20 34 2D 39 2D 32 ; 璁板事鏈? 4-9-2
00000100h: 30 31 33 22 0A 40 50 4A 4C 20 53 45 54 20 53 54 ; 013".@PJL SET ST
00000110h: 52 49 4E 47 43 4F 44 45 53 45 54 3D 55 54 46 38 ; RINGCODESET=UTF8
00000120h: 0A 40 50 4A 4C 20 4A 4F 42 20 4E 41 4D 45 3D 22 ; .@PJL JOB NAME="
00000130h: 69 6D 61 67 65 32 2E 6C 6F 67 20 2D 20 E8 AE B0 ; image2.log - 璁?
00000140h: E4 BA 8B E6 9C AC 22 0A 40 50 4A 4C 20 43 4F 4D ; 浜嬬潃".@PJL COM
00000150h: 4D 45 4E 54 3D 22 68 70 20 4C 61 73 65 72 4A 65 ; MENT="hp LaserJe
00000160h: 74 20 31 31 36 30 20 28 36 31 2E 36 33 2E 34 36 ; t 1160 (61.63.46
00000170h: 31 2E 34 32 29 3B 20 57 69 6E 64 6F 77 73 20 37 ; 1.42); Windows 7
00000180h: 20 55 6C 74 69 6D 61 74 65 20 36 2E 31 2E 37 36 ;  Ultimate 6.1.76
00000190h: 30 30 2E 31 3B 20 55 6E 69 64 72 76 20 30 2E 33 ; 00.1; Unidrv 0.3
000001a0h: 2E 37 36 30 30 2E 31 36 33 38 35 22 0A 40 50 4A ; .7600.16385".@PJ
000001b0h: 4C 20 43 4F 4D 4D 45 4E 54 3D 22 55 73 65 72 6E ; L COMMENT="Usern
```

**图 3　传递给打印监视器的数据流**

可以看出这是一个打印控制语言写成数据流，应该属于 PCL 类的，PCL 是 HP 发明的由于它不收费所以应用的比较多一些，但其版本比较多。这个数据流会传递给打印监视器由它来传给打印机。WritePrinter 一次调用如果传递不完全连续调用几次直到传递数据流的所有内容。数据流传递结束后会调用本地打印提供的 SplEndDocPrinter 函数。下面是展示了这个函数的调用栈信息，还可以看到一直跟踪的那个句柄值。综合前两次的栈信息可以看出打印过程是在打印处理器的 PrintDocumentOnPrintProcessor 中完成的：

ChildEBP RetAddr  Args to Child
063def30 00db9436 05f3cba8 063def4c 00da92ca localspl!
SplEndDocPrinter (FPO: [Non-Fpo])
063def3c 00da92ca 05f20620 003991f4 063def94 spoolsv!
EndDocPrinter+0x1d (FPO: [Non-Fpo])
063def4c 738bcd32 05f20620 00000000 7b03ddcb spoolsv!
YEndDocPrinter+0x22 (FPO: [Non-Fpo])
063def94 7640591b 003991f4 06496450 063defc0 winspool!
EndDocPrinter+0xfe (FPO: [Non-Fpo])
063defa4 763f5220 06830290 003991f4 00000000 GDI32!
EndDocPrinterEx+0x5c (FPO: [Non-Fpo])
063defc0 763f9c00 b42103eb a3ccdfee 05f3e188 GDI32!
EndDoc+0xa2 (FPO: [Non-Fpo])
063deff4 73785503 003260f8 7b0f985c 00000001 GDI32!
GdiEndDocEMF+0x37 (FPO: [Non-Fpo])
063df058 737859a9 003260f8 b42103eb 063df0a4 winprint!
BPrintEMFJobNow+0x1be (FPO: [Non-Fpo])
063df18c 737834db 05f3e188 00326134 063df1bc winprint!
PrintEMFJob+0x48f (FPO: [Non-Fpo])
063df19c 73d542b7 05f3e188 063df30c 00000000 winprint!
PrintDocumentOnPrintProcessor+0x3c (FPO: [Non-Fpo])
063df1bc 6149c35c 0663737c 063df30c 00b06a4a
PrintIsolationProxy!sandbox::PrintProcessor::PrintDocThrough
PrintProcessor+0x49 (FPO: [Non-Fpo])
063df228 6149b052 00ad6744 063df30c 00b06a4a localspl!
sandbox::PrintProcessorExecuteObserver::
PrintDocThroughPrintProcessor+0x84 (FPO: [Non-Fpo])
063df244 61499d20 00ad6744 063df30c 00b06a4a localspl!
sandbox::PrintProcessorAdapterImpl::
PrintDocumentOnPrintProcessor+0x27 (FPO: [Non-Fpo])
063df270 61460400 00aa3960 063df30c 0018c938 localspl!
sandbox::PrintProcessorAdapter::
PrintDocumentOnPrintProcessor+0x77 (FPO: [Non-Fpo])
063df950 614615fe 00aa4f20 063df970 00000000 localspl!
PrintDocumentThruPrintProcessor+0x3bd (FPO: [Non-Fpo])
063dfbb0 76161174 00aa4f20 063dfbfc 77d1b3f5 localspl!
PortThread+0x41c (FPO: [Non-Fpo])
063dfbbc 77d1b3f5 00aa4f20 70896cd8 00000000 kernel32!
BaseThreadInitThunk+0xe (FPO: [Non-Fpo])
063dfbfc 77d1b3c8 614611e2 00aa4f20 00000000 ntdll!
_RtlUserThreadStart+0x70 (FPO: [Non-Fpo])
063dfc14 00000000 614611e2 00aa4f20 00000000 ntdll!
_RtlUserThreadStart+0x1b (FPO: [Non-Fpo])

## 3　内容审计方案

通过第二节的内容可以了解到在本地打印中存在两个比较明显的监视点。一个是可以在打印处理器的 PrintDocumentOnPrintProcessor 接口中截获 EMF 等格式的数据流，另一个是在本地打印提供者的 SplWriterPrinter 接口中截获 PCL 等格式的数据流。前者称为打印处理器方案，后者称为打印提供者方案。

### 3.1　打印处理器方案

打印处理器可以有多个，与具体的打印机相关。如果打印机接收的打印控制语言微软提供的通用打印处理器无法转译时

就需要厂商自己开发一个。因此在这个方案中第一步要实现一个打印处理器枚举的功能；第二步就是解决截获数据流的问题。

第一个问题解决多个打印处理器枚举的问题，所有打印处理器的信息都在注册表的HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Environments\Windows NT x86\Print Processors 下面。只要对这个地方子键进行枚举即可。得到所有的打印处理器名称和 DLL 名后分别针对 HOOK 处理即可。

第二个问题如何解决截获数据流是问题。打印处理器必须提供一组接口，它们分别是：ControlPrintProcessor、EnumPrintProcessorDatatypes、GetPrintProcessorCapabilities、ClosePrintProcessor、OpenPrintProcessor、PrintDocumentOnPrintProcessor。它们一共 6 个这是一个打印处理器必须导出的。但是有用的只有 ClosePrintProcessor、OpenPrintProcessor、PrintDocumentOnPrintProcessor 这 3 个函数。因此只要 HOOK 这 3 个函数就可以了，同时因为打印处理器是被假脱机服务调用 LoadLibrary 函数加载的，所以要用 INLINEHOOK 才有用，另外还有一个问题才是采用 INLINEHOOK 的一个重要原因，下面会说明。在 OpenPrintProcessor 调用时新建一个审计上下文块用于保存打印任务相关的信息，在 PrintDocumentOnPrintProcessor 调用时把数据另存一份，比如可以写到一个临时文件中，ClosePrintProcessor 调用时表明这次打印结束，可以提交审计内容，同时释放本次打印审计上下文块。

因为可能存在多个打印处理器，每一个打印处理都导出的是相同的函数。这样的话就需要为每一个打印处理器写一份处理代码。因为目标机器上的打印处理器个数又不是固定的，所以这个问题比较难解决。但是解决方法还是有的，可以准备一段特殊的 SHALLCODE 模板。用它来做跳板调用审计功能代码，审计功能返回后再跳回处理器原来的接口。对每一个打印处理器进 HOOK 时先申请一块内存，再把 SHALLCODE 代码填充上让函数执行时跳到这块内存上。这里提供一种 SHALLCODE 的简单布局方案。如图 4 所示。

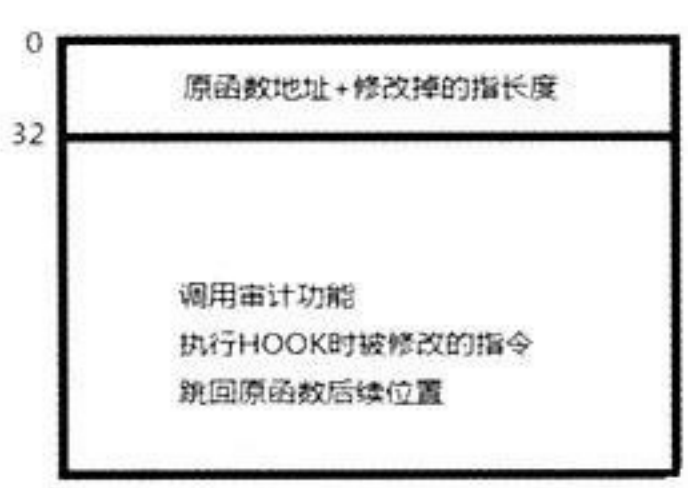



图 4 SHALLCODE 的简单布局

一般情况在这里截获到的 SPL 格式的文件要从中剥离出来 EMF 文件来对打印内容审计才有用。如果打印的是多个页面则，每一个页面对应一个 EMF 文件，但是它们都被连在一个 SPL 文件流中，因此如果打印的是多个页面，那剥离时则要存成多个 EMF 文件。如果打印的内容是文本还可以在 EMF 文件中解析出打印的文本内容。如果是图片就要麻烦一些了。

### 3.2 打印提供者方案

打印提供者在系统中也是多个的，但因为针对的是本地打印审计，而本地打印提供者只有一个，那这就不是什么问题了。打印提供者要提供的接口是比较多的。这里不能一一列举了，但只要关注两个就可以了。打印提供者需要对假脱机服务提供一组接口，这组接口是通过 InitializePrintProvidor 函数导出的。函数参数提供一个 PRINTPROVIDOR 结构。它的成员一组函数指针包括打印提供者必须要导出的函数，要通过这个 InitializePrintProvidor 函数来得到 WriterPrinter 和 EndDocPrinter 地址，再对其进行 INLINE HOOK。

审计功能的实现是在 WriterPrinter 中把向打印机写的数据流转储到一个文件中，当 EndDocPrinter 调用时说明本次打印结束，把转储的数据做为审计数据保存起来。在这里截获的数据都的 RAW 数据。大部分打印采用的 HP 的 PCL 中的一种。因此需要程序提供 PCL 的解析能力，把相应的 PCL 文件转成位图或文本。这样就能实现打印审计了。

### 3.3 方案对比

两种方案在实现的功能的效果上差距不大，差距主要体现在实现成本上。打印处理器方案主要的难点在 INLINE HOOK 的实现上，因为要对其多个打印处理器同时 HOOK，因此需要一段特殊的 SHALLCODE 来完成，而且 64 位与 32 位需要区别对待其稳定性的控制上要难一些。在打印提供者方案中的难点是对多种 RAW 格式的文件进行解析。因为 PCL 格式版本众多而没有功能全且免费的解析库，所以要麻烦很多。然而有些非 PCL 版本文件格式本身就是收费的。相对来说 EMF 的解析就简单多了而且格式是开放的。

## 4 结语

介绍了两种方案的利弊。除此之外，还有一些方案如代理端口监视器的方案，但这种方案截获的数据与打印提供者方案相同，其困难也差不多。还有虚拟打印机的方案，编写一个虚拟打印统一先由虚拟打印机转成相关格式后再打印，在这个过程中很难实现自动化。这样会改变用户的一些使用习惯，用户体验较差。如上探讨的只是一种技术方案，细节实现可能有很多种。

（收稿日期：2013-04-24）

# 科学数据打包与分发技术

闫海忠　杨汝龙

**摘　要**：生态的调查研究和实验过程常常会涉及到诸多数据，这些数据往往类型多样，数据量极大，数据获取相当不易，对数据管理和使用提出了更高的要求。利用 Wise Install System9.02 作为这类数据管理和分发的工具，面对纷扰繁杂的各种各样数据，能够对科学数据进行组织和管理，打包与分发，实现更为有效管理和安全使用。

**关键词**：科学数据；打包与分发；Wise installation 工具；生态数据

## 1　Wise install system9.02 简介

非常著名的安装程序制作工具，它提供脚本编辑方式及众多应有尽有的安装选项，堪比专业级的安装程序制作软件。Wise 支持创建一个独立的可执行文件以便于在线发布程序，也能够支持多磁盘，并且支持网上（HTTP 和 FTP 方式）分发，支持调用外部 DLL、EXE 等，灵活的脚本控制，根据多年数据打包的经验，较之其他类型的软件，它具有体积小，安装使用方便，打包分发安全可靠。以下简称 Wise902。

## 2　数据和软件准备

### 2.1　数据准备

生态数据（ecological data）以反映生态信息的属性为测量指标而测得的数据。生态数据是以植被数量分析为基础的各类信息，一般包括两大类型：

一类是反映群落组成、结构关系的植物区系组成数据，这些数据是反映群落成员特征的一些定量和定性的属性数据，即数量数据和二元数据。

另一类是群落的环境组成数据，包括各种环境因子的测量指标。

所以，生态数据涉及不少类型的数据，在本例中有遥感数据、空间地理数据、视频文件、录音文件，调查表格和其他研究资料等。逐一将它们准备好放置在相应的计算机磁盘中备用。

### 2.2　工具软件

安装后的 Wise902 提供了 Installation Expert 和 Script Editor 两种控制打包程序的方式。推荐读者使用 Installation Expert 模式，它是一种向导的模式，以这种模式为主，在向导模式的引领下能够更快更好完成一个复杂的数据打包任务。Script Editor 模式是基于脚本，脚本语法有点像 Basic 语言。可以在某些特殊的数据使用时再应用它（例子中分发安装后执行外部程序部分有介绍），它左边有一个列表专门提供可以供调用的脚本语句，需要时选择调用。

## 3　数据打包

数据打包即数据和应用装配过程，这个过程在 Wise902 中变得相当容易。下面就来实现这类数据的打包实践。

### 3.1　建立工程文件和设置

启动 Wise902 后，新建一个工程文件，命名为：“科学数据.WSE.”，并在“安装标题中”填入：“生态环境数据的打包与分发”，“默认目录”一栏填入：“生态数据”，将“默认目录放置在‘Programming Files’的目录下”勾选。如图 1 所示。

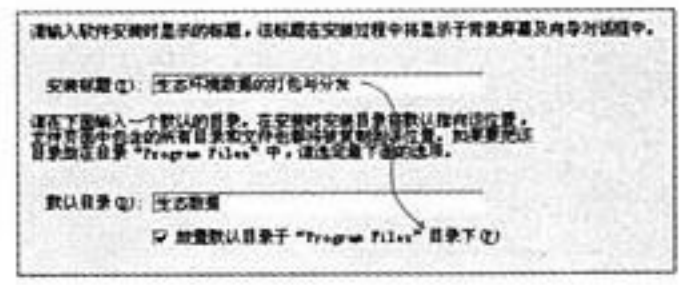


**图 1　数据包标题和安装默认目录的设置**

### 3.2　添加组件和命名

Wise902 提供数据打包的分组打包功能，利用该组件功能在使用时可以将数据分不同类型进行分装，方便将不同类型的数据源进行打包和管理。在本例中，所有生态数据按照实际所需，分为基础数据、专题数据、气象数据、地理空间、遥感、群落样地、群落样方、社会经济、生态计算（外部计算程序）以及相关的环境录像和音频数据等，共 11 种数据类型。它们将通过 Wise Installation 的组件装配功能创建对应的数据类型名称。具体步骤如下：在方案定义部分点击“组件”按钮，之后，在弹出的组件对话框中再点击“添加”按钮，在弹出的组件详情对话框中，填入相应组件名称，并勾选“默认安装组件(I)”选项即可，如图 2 所示。重复此步骤，逐一将上述 11 个

数据类型组件添加完毕，形成了数据包所有数据栏目，以便稍后所有生态数据分装进来。

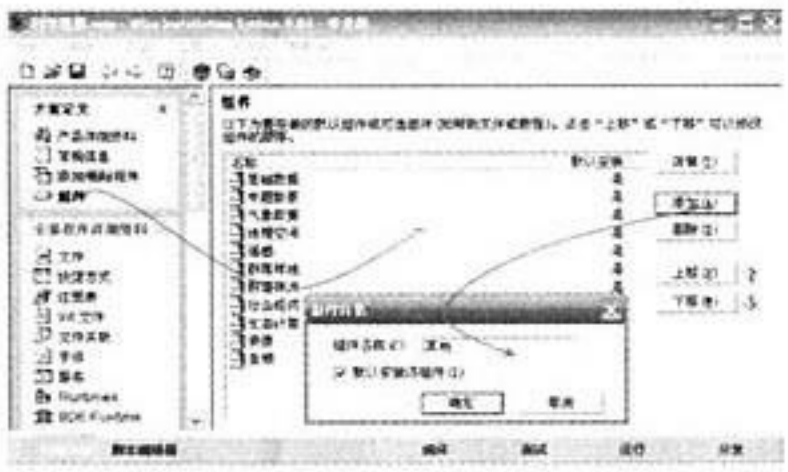


图 2 数据分装组件的添加和定义

## 3.3 数据源文件加入

Wise902 提供将现有磁盘中的数据文件加入到当前工程应用中来。步骤如下：在安装程序详细资料页面中选择点击“文件”，弹出文件选择，并加入对话框，通过它可以按照所创建的数据分装组件一一地将已经准备好数据添加到包中来，本例中将 1 号样地所涉及到 11 类数据文件全部按要求加到了工程里面，如图 3 所示。

（需要注意的在添加目录区操作时，新建目录和添加文件最好添加一个目录就将所要文件添加进来，否则 Wise902 系统会出错，其他版本有无问题暂不知道）

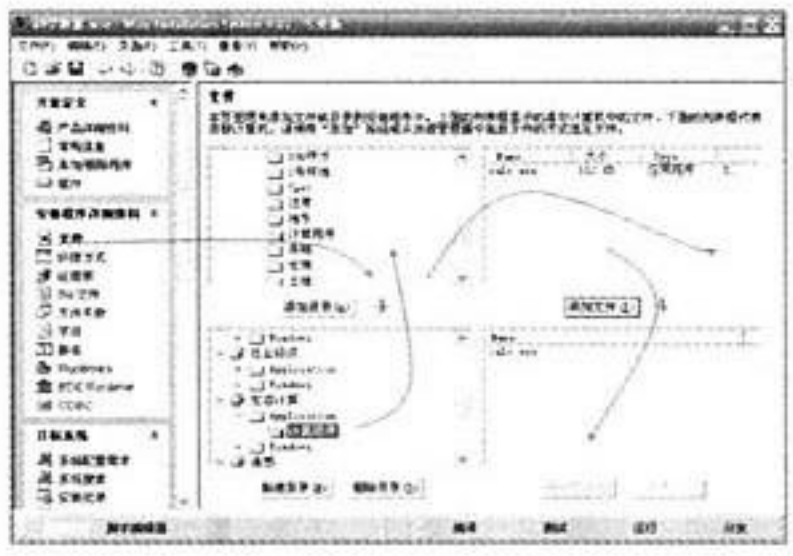


图 3 数据源文件的加入

## 3.4 添加快捷方式

由于本例中有计算程序，可以使用专家模式的快捷方式页来向目标电脑上的桌面和开始菜单上添加快捷方式。要在安装过程中添加快捷方式：

（1） 点击“快捷方式”，弹出快捷方式，填入相应的内容，然后单击“添加”按钮，如图 4 所示。

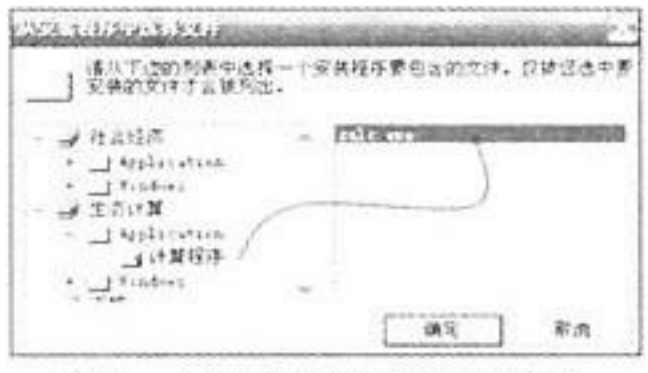


图 4 添加计算程序快捷方式

（2） 从安装对话框中选择文件，在左边选择包含你想要与之关联的文件类型的程序文件的目录，在右边选择要关联的快捷方式的文件。

（3） 点击“确定”，然后在快捷方式的详细资料对话框中编辑快捷方式的详细信息。

## 3.5 添加注册表键和键值

作为一个专业安装包有时候需要想 Windows 注册表添加相应的包特征信息，可以使用专家模式的注册表页来制定要在目标计算机上添加或编辑的注册表项。上面的两个列表框显示了本地计算机上的注册表键和键值。下面的两个列表框显示将要在目标计算机上添加的键和键值，如图 5 所示。

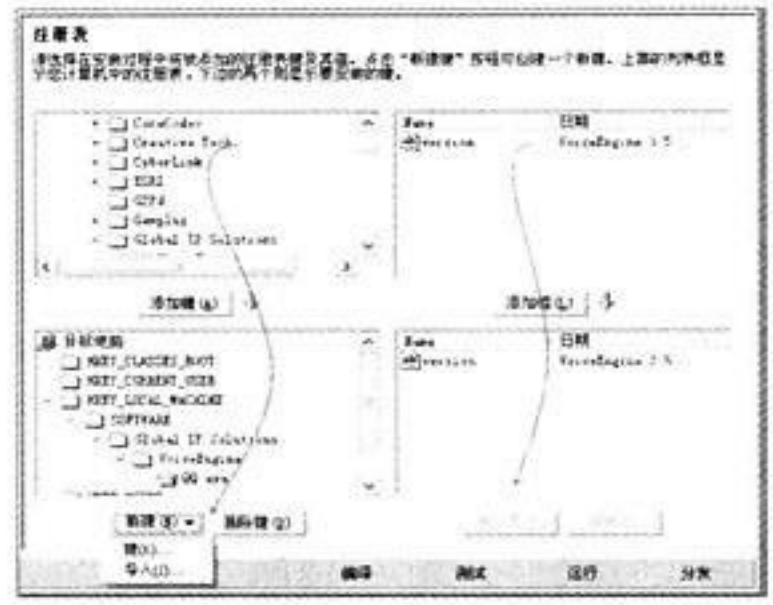


图 5 添加注册表键和键值

“添加键”按钮可以复制一个完整的注册表键，“添加值”按钮可以复制键值，“新建”按钮可以通过导入一个注册表文件来创建一个新的注册表项。

要添加一个注册表项：

（1） 在下面左侧的列表框中单击选择想要添加的键值。

（2） 单击“新建”按钮然后从下拉列表中选择相应的键。

（3） 在这册表项设置对话框中配置注册表值。按 F1 启动帮助。

## 3.6 添加关联文件

生态数据中有的要用某一类程序才能打开，在专家模式下使用关联文件页可以配置关联一个文件的应用程序用来打开这个类型的文件。要为一个文件类型配置一个关联程序：

（1） 在关联文件页，单击“添加”按钮，弹出文件选择对话框，如图 6 所示。

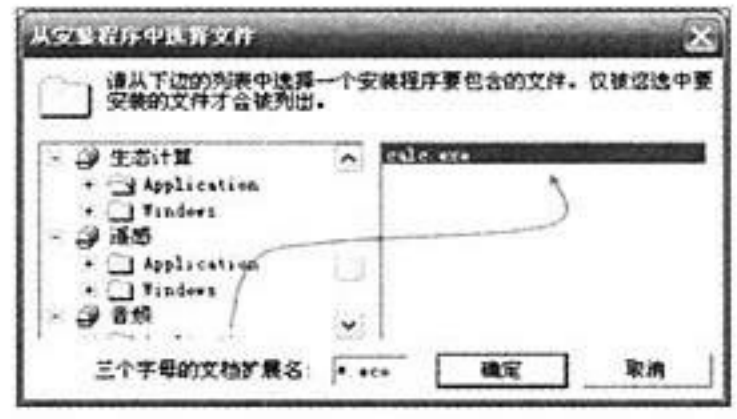


图 6 文件关联操作

(2) 从安装对话话框中选择文件，在左侧选择包含要关联的文件类型的可执行文件的目录，右侧为要关联的文件。

(3) 在对话框的底部，数据 3 个字符的扩展名来标识关联的文件类型。

(4) 单击“确定”。

要编辑一个文件关联的设置，双击文件关联页中的项目即可。

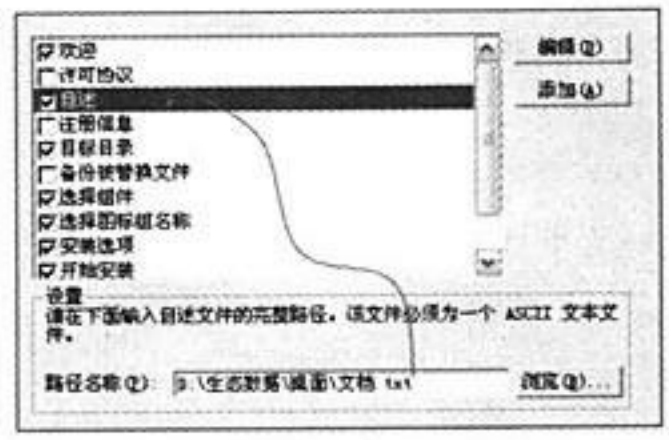



图 8 安装程序对话框选择与编辑

## 3.7 指定系统配置需求

通过专家模式中的“系统配置需求”页，可以指定安装程序运行的最低软硬件需求，同时可以设置如果目标电脑的不满足最低需求时出现的警告信息。

这里有一个例子用来制定在 Windows XP 下安装程序最低的操作系统需求。

(1) 在目标系统需求页，双击 “Windows NT 版本”,弹出配置对话框，如图 7 所示。

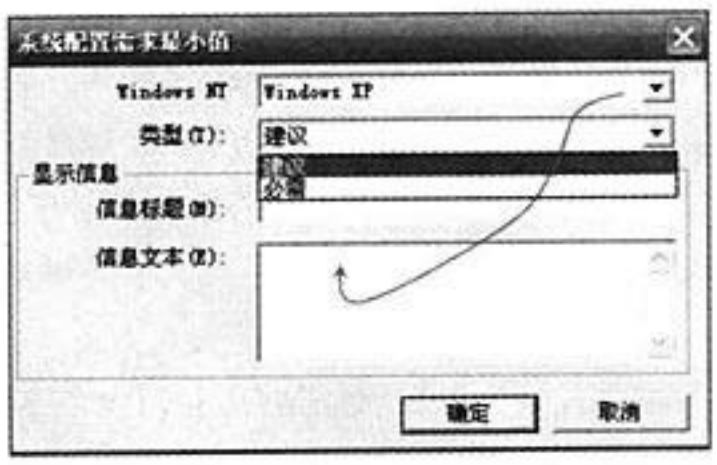



图 7 安装系统需求配置

(2) 在最低系统需求对话框中找到 Windows 版本下拉列表，选择 Windows XP。

(3) 从“类型”下拉列表中选择“建议”或者“必需”。如果选择的是“必需”，而目标系统不满足系统，则安装程序将终止安装。

(4) 为消息对话框输入标题和内容，如果目标电脑低于 Windows XP 或更高的操作系统，那么会弹出这个消息对话框。

(5) 单击“确定”。

## 3.8 选择安装对话框

通过点击在专家模式用户界面页面的“对话框”项，选择合适的安装时出现的对话框界面，可以指定在安装期间出现的对话框样式。要查看选择的对话框样式，可以勾选某个对话框然后双击样式名字，并即将打开自定义对话框编辑器。

下面是如何添加一个“自述”对话框的例子：

(1) 在对话框页，标记“自述文件”选择框并双击。如图 8 所示。

(2) 在路径名称区，输入要使用的自述文本文件的路径名称。

(3) 需要修改对话框样式可点击“编辑”按钮进行。

## 3.9 BDE 配置

本实例中生态计算程序设计部分数据库文件的使用，所以需要针对它们完成数据库引擎 BDE 的设置，通过点击在专家模式中“BDE Runtime”页，弹出数据库引擎配置对话框，如图 9 所示。要实现 BDE 配置：

(1) 在 BDE 安装类型 (P) 处，选择部分 BDE 32 安装选项。

(2) 在 BDE 32 子集页处，勾选 SQL,Paradox 和 DBASE 选项。

(3) 如需要添加本机中的 BDE 别名，点击“添加”按钮。

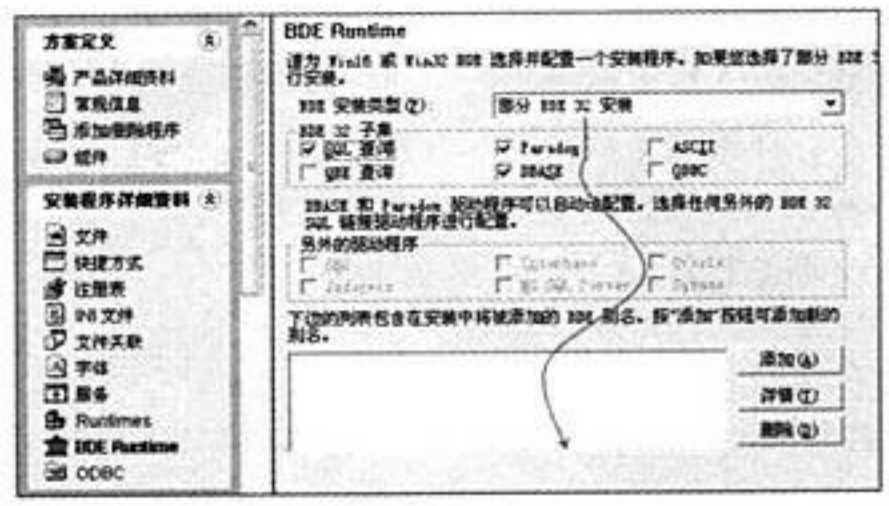



图 9 数据库引擎 BDE 配置

## 3.10 安装密码

从数据安全的角度，有必要给所形成的安装包设置权限。Wise 902 提供了这一功能，在安装选项页面，选择并点击密码弹出密码设计对话框，如图 10 所示。要实现安装密码的设置：

(1) 选择在“所有安装程序使用单一密码”，并设置所需要的密码内容。

(2) 如果需要类似专业软件安装系列号，选择“使用个别的序列号作为密码”设置。

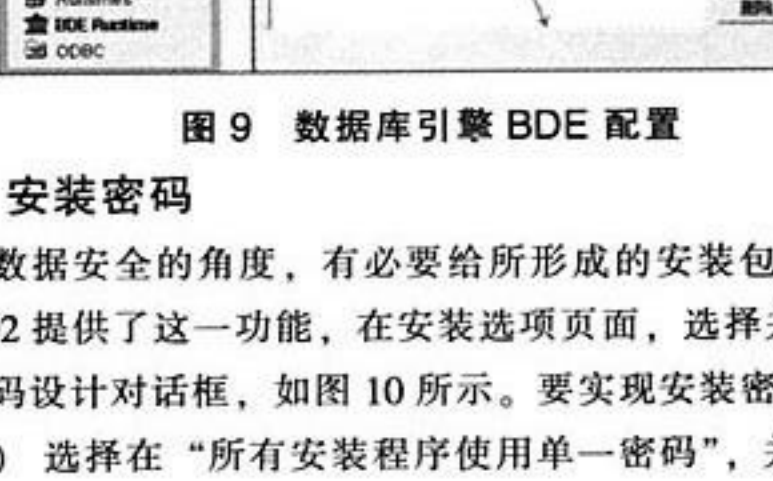



图 10 安装包密码设定

## 3.11 分发安装后执行外部程序

有时候当数据包安装在目标计算机后，需要执行某个外部

应用程序，本实例中就是设计当安装包安装解压后自动执行包中的生态计算程序（calc.exe）。

（1）在用户界面页面的安装对话框中，勾选“安装选项”对话框，使这一界面在安装过程中出现以便选择“安装完成后开始执行程序”，一旦选择了此项，系统将自动执行设置好的外部可执行文件，如图 11 所示。

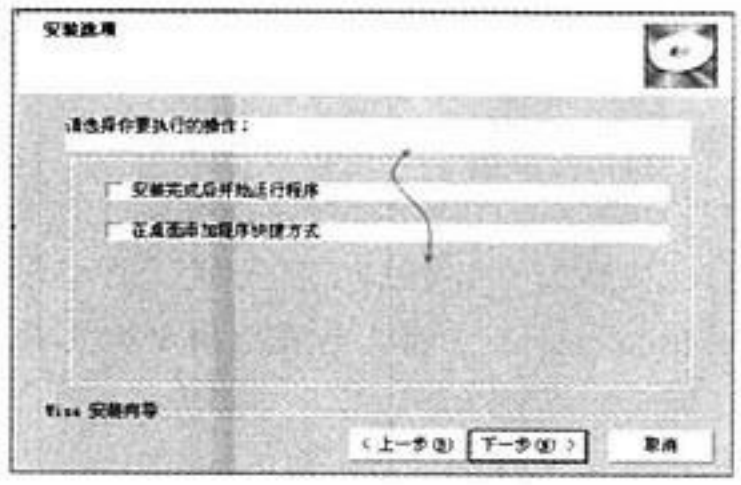



图 11 数据包分发安装后执行程序

（2）通过双击在脚本编辑器页面的“执行程序”项，在弹出的执行程序设置的对话框界中进行程序文件浏览和选定，可以选择任何打入包中程序文件，同时脚本部分内容也将自行加入或更新，即增加了新的脚本内容：

Rem 在这里设定退出安装要运行的程序：

执行%MAINDIR%\计算程序\calc.exe

If RUN 等于 "A" then

执行%MAINDIR%\计算程序\calc.exe

（其中 RUN 变量，勾选时：RUN = "A"，不选时：RUN = "B"）

如图 12 所示。

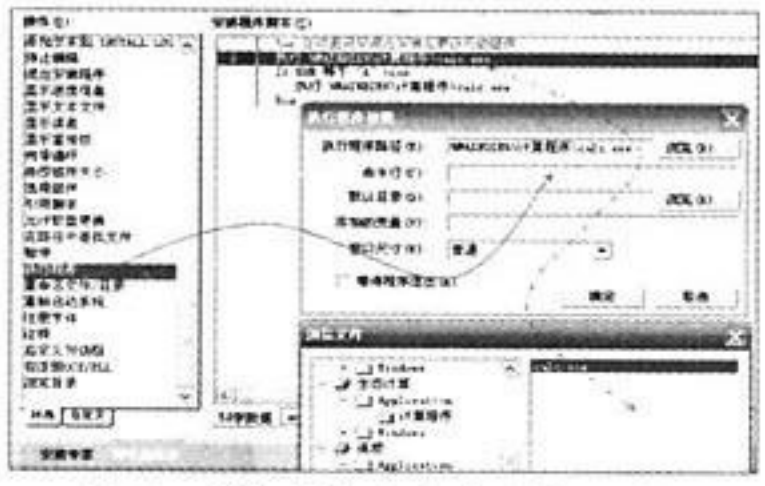


图 12 外部执行程序的浏览和选择

至此，数据的打包和设置已经结束，需要将该数据包工程文件（科学数据打包与分发.wse）保存。

## 4 数据分发

数据分发与数据打包过程基本相反，是将所装配的数据和应用程序分装到不同的介质上，并通过安装程序将包中所有数据和程序按打包时的要求部署到目标计算机中。

### 4.1 分发介质

介质指存放数据包的物理设备，在 Wise 902 中介质可以是多种类型的，分发前可以进行选择。要实现安装包分发介质的选择，通过点击在专家模式编译选项页面的“介质”项实现，如图 13 所示。


图 13 分发介质设置

（1）单一文件安装程序：创建一个独立的磁盘文件，该文件与工程文件同名。

（2）基于介质的安装程序：该选项将数据包的内容分割为适合的介质类型文件，有多个文件组成（*.W0x，x>2），保证数据能够存放到相应介质中。

### 4.2 编译安装程序

一旦完成了创建或修改一个安装程序，可以通过位于主窗口右下方的编译，测试和运行按钮来进行调试。

（1）点击“编译”，编译所创建的安装程序，在工程文件位置生成可执行安装包程序，如本例的科学数据打包与分发.exe。

（2）点击“测试”，模拟安装过程，但是不对系统做任何修改。

（3）点击“运行”，编译和实际运行所生成的安装程序。

图 14 是该安装包程序执行过程的两个数据分发的交互界面，通过操作该程序文件分装过程，可以看出 Wise 902 无论是数据装配，还是分发与安装在功能和操作上都是非常专业和方便的。

a.分发可选数据　　b.安装结束后执行动作

图 14 数据安装包程序文件的运行

## 5 结语

利用 Wise 902 和生态数据进行打包和分发全部过程已介绍完成，读者可以体会到该工具的专业性和简便性。尤其是在应用程序数据库数据库文件，利用它进行打包和分发，安装部署均显得心应手，不像其他的安装制作工具使用过程过于繁杂。经常做数据打包和分发的人会发现，实际上有不少都是用 Wise Installation System 完成的，而且数据的安全性也是有保证的。

（收稿日期：2013-07-04）

## TROUBLESHOOTING OF PROGRAM

**Q** 如何实现 C/S 模式数据库应用系统升级

**A** 在企业使用的 C/S 模式数据库应用系统中，随着用户需求的不断变更和提升，应用系统需要修正与完善，但系统的更新与维护工作比较频繁。通过长期应用实践，找到一种非常简单实用的升级方案，就可以让用户随时自动升级到最新版本的系统，同时可大大降低维护工作。

**(1) 设计原理**

首先通过应用程序从数据库中获得需要升级的需求，然后关闭应用程序自己，运行升级程序从数据库中得到新版应用程序，并覆盖原应用程序，最后关闭自动升级程序，并运行应用程序。升级架构图与流程图如图 1 和图 2 所示。

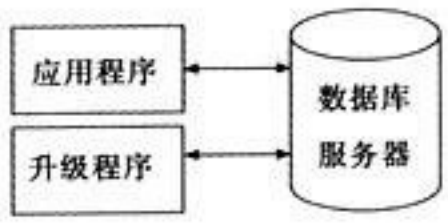



图 1 系统升级架构

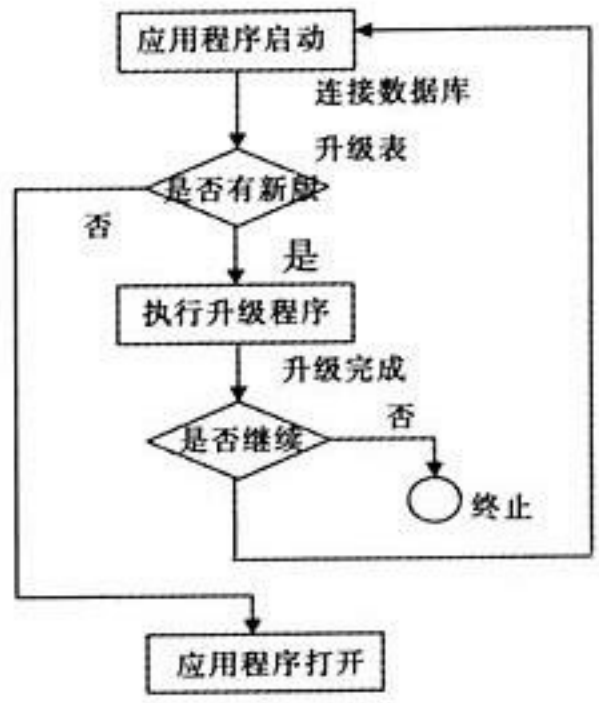



图 2 系统流程

**(2) 编程实现**

首先在数据库中建一个升级表，以下是以 Oracle 数据库创建表的例子，其他数据库需要更换对应的字段类型。

```
create table HRUPGRADE
(
  VERID  NUMBER not null, //版本号
  UPDATE_DATE DATE,   //应用系统更新日期
  UPDATE_ZGBH VARCHAR2(8),   //应用系统更新操作人
  PRO   LONG RAW    //大字段存放应用程序
);
```

注：这里用的 Delphi；的 BDE 连接，BDE 驱动程序对 Oracle BLOB 字段支持不是很好，下载下来的应用程序会增加一个空白多余的字节，所以字段采用的 LONG RAW 类型。

然后 C/S 方式数据库应用系统增加检测当前最新版本号功能，以下是 Delphi 为应用程序开发工具的例子。

1）增加一个检测当前最新版本号的窗体（Tfrm_Upgrade Check），添加连接数据控件：Database1，Query2，并设置好数据源连接到数据表 HRUPGRADE，如图 3 所示。


图 3 检测当前最新版本号

单元代码如下：

```
unit UpgradeCheck;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs, ExtCtrls,  Db, DBTables;
type
  Tfrm_UpgradeCheck = class(TForm)
    Database1: TDatabase;
    Query2: TQuery;
  private
    { Private declarations }
    FServerVER:Integer;//服务器版本信息；
    FLocateVER:Integer;//自身版本信息
  public
    { Public declarations }
    procedure GetServerVER;
  end;
var
  frm_UpgradeCheck: Tfrm_UpgradeCheck;
  function callUpgradecCheck:Boolean;
implementation
uses TgDialogs;
{$R *.DFM}
function callUpgradecCheck:Boolean;
begin
  with Tfrm_UpgradeCheck.Create(application) do
  try
    result := False;
    GetServerVER;
    FLocateVER := 7;   //设置此应用程序自身版本号
    if FServerVER<>FLocateVER then
      result := true;
  finally
    free;
  end;
end;
{ Tfrm_UpgradecCheck }
procedure Tfrm_UpgradeCheck.GetServerVER;
begin
  try
    FServerVER := 0;
      Database1.Connected := True;
    Query2.SQL.Clear;
```

```
    Query2.SQL.Text := 'select max(VERID) from hrupgrade';
    Query2.Active := True;
      FServerVER:=Query2.Fields[0].AsInteger; //获得最新的版
//本号
  finally
    Query2.Active := False;
    Database1.Connected := False;
   if FServerVER = 0 then FServerVER :=  FLocateVER;
  end;
end;
```

2）应用程序启动前增加版本检查

```
...
   splashform.show;
   SplashForm.Update;
            if callUpgradecCheck then//是否有升级程序,调用
//UpgradeCheck 下过程,True 有新版本
   begin
      Fn:= ExtractFilePath(ParamStr(0))+'HRUPGRADE.EXE';
     // HRUPGRADE.EXE 升级程序升级更新 HR 程序
      FillChar(TSI, SizeOf(TSI), 0);
      TSI.CB := SizeOf(TSI);
      IF CreateProcess  (PChar (Fn), NIL, NIL, NIL, False,
DETACHED_PROCESS, NIL, NIL, TSI, TPI) then
        exit
      else
      begin
        messagebeep(0);
        Application.MessageBox('不能完成文件拷贝 '+' 请手
工拷贝! ','Copy Error! ',1);
        Exit;
      end;
   end;
   ...
```

3）创建升级程序工程，添加连接数据库的控件（如图 4 所示），因为一般应用数据库系统可执行文件都比较大，TQuery 控件下载的话，缓存不够，可放一个 TTable 控件，用来下载可执行文件：

```
[Query1]
SQL: select * from hrupgrade order by verid desc
[DataSource1]
DataSet: Query1;
[Table1]
MasterSource: DataSource1
MasterFields:VERID
IndexFieldNames:VERID
TableName: hrupgrade
procedure TfrmUpGrade.UpdateHR;
begin
  try
    Database1.Connected := True;
    Query1.Active := True;
    Table1.Active := True;
    Animate1.Visible := true;
    Animate1.Active := true;
    if Table1.active and (Table1PRO.asstring <> '') then
    begin
       TBlobField    (Table1PRO).SaveToFile   (ExtractFilePath
(Application.EXEName) + '\HR.EXE');
    end;
     Query1.Active := False;
     Database1.Connected := False;
    if Application.MessageBox('人力资源管理系统已成功升级,
要继续运行吗?','提示', MB_ICONQUESTION +
MB_OKCANCEL+MB_DEFBUTTON1) = IDOK then
        RunHRSystem('HR.EXE');
    Application.Terminate;
  finally
    Query1.Active := False;
    Database1.Connected := False;
  end;
end;
```

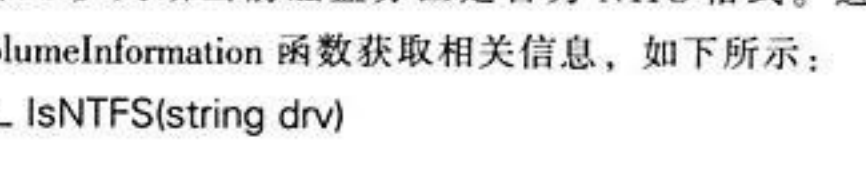



图 4　升级系统界面

目前 B/S 方式应用系统集中发布方便，但客户端浏览器经常出故障，维护也困难，相比 C/S 方式应用系统通过构建一个简单实用升级程序，却可以大大降低维护的难度。

（作者：范士奇）

**Q　如何编程实现读取 USN 日志**

**A**　USN 日志是 Windows 的 NTFS 磁盘格式下的特有数据库，每个卷（即磁盘盘符）都有独立的 USN 日志，它记录了所有文件和目录的变化情况。操作 USN 日志一般分为 5 个步骤。

第 1 步判断当前磁盘分区是否为 NTFS 格式。这是通过 GetVolumeInformation 函数获取相关信息，如下所示：

```
BOOL IsNTFS(string drv)
{
      TCHAR fmt[MAX_PATH]={0};
      BOOL bRet(0);
      bRet =GetVolumeInformation  (drv.c_str (),0,0,0,0,0,fmt,
MAX_PATH);
      return((bRet&&0==lstrcmp("NTFS",fmt))?1:0);
}
```

其中第一个参数是传入的分区名，比如写成 d: 这样，其他参数基本可以忽略。

第 2 步获取磁盘驱动器句柄，以便后续操作使用，如下

## TROUBLESHOOTING OF PROGRAM

所示：

```
HANDLE lamAdmin(string root)
{
    HANDLE hVol(HANDLE(-1));
        hVol =CreateFile (root.c_str ),GENERIC_READ |
GENERIC_WRITE,
            FILE_SHARE_READ |FILE_SHARE_WRITE,0,
OPEN_EXISTING,0,0);
        return ((hVol! =INVALID_HANDLE_VALUE)?hVol:
(HANDLE(-1)));
}
```

这里需要注意的是，必须设置 OPEN_EXISTING 标识，另外共享属性里需要设置 FILE_SHARE_WRITE 标识，否则调用会失败。而且在打开磁盘盘符时，即 CreateFile 的第一个参数，需要管理员权限，必须这样设置：\\\\.\\d:才可以。

第 3 步初始化 USN 日志文件，需要调用 DeviceIoControl 函数，使用其中的 FSCTL_CREATE_USN_JOURNAL 指令码，如下所示：

```
BOOL InitUSN(HANDLE hVol)
{
    CREATE_USN_JOURNAL_DATA cujd={0};
    DWORD cbRet(0);
    BOOL bRet(0);
    cujd.AllocationDelta=cujd.MaximumSize=0;
                        bRet =DeviceIoControl (hVol,
FSCTL_CREATE_USN_JOURNAL,&cujd,
        sizeof(cujd),0,0,&cbRet,0);
    return(bRet);
}
```

现在已经成功创建 USN 日志，当然若以前曾经生成过日志信息且未使用 FSCTL_DELETE_USN_JOURNAL 指令码删除，也可以免去此步骤。

第 4 步枚举所有日志条目，使用的 API 与前步相同，只是指令码换成 FSCTL_ENUM_USN_DATA 即可，如下所示：

```
voidEnumU SN (HANDLE hVol,hash_multimap <string,
FileGuid>&hms,
    hash_multimap<UINT64,AuxIdx>&hmu)
{
    USN_JOURNAL_DATA qujd={0};
    DWORD cbRet(0);
    BOOL bRet(0);
bRet=DeviceIoControl(hVol,FSCTL_QUERY_USN_JOURNAL,
        0,0,&qujd,sizeof(qujd),&cbRet,0);
    if(bRet)
    {
        MFT_ENUM_DATA med={0};
        med.HighUsn=qujd.NextUsn;
        med.LowUsn=med.StartFileReferenceNumber=0;
        vector<BYTE>buf(BUF_SIZE);
            PUSN_RECORD pUsnRec(0);
for(;DeviceIoControl(hVol,FSCTL_ENUM_USN_DATA,&med,
                sizeof(med),&buf[0],buf.size(),&cbRet,0);)
            {
    pUsnRec=PUSN_RECORD(&buf[0]+sizeof(USN));
                for(cbRet-=sizeof(USN);cbRet>0;)
                {
                    TCHAR name[MAX_PATH]={0};
                    intdalen(pUsnRec->FileNameLength);
                    WideCharToMultiByte (CP_OEMCP,0,
pUsnRec->FileName,dalen/2,name,dalen,0,0);
                    DWORD attr(pUsnRec->FileAttributes);
    UINT64 fid(pUsnRec->FileReferenceNumber);
    UINT64 pid(pUsnRec->ParentFileReferenceNumber);
                    FileGuidfg={fid,pid,attr};
                    hms.insert(make_pair(string(name),fg));
                    AuxIdxai={string(name),pid};
                    hmu.insert(make_pair(fid,ai));
                    cbRet-=pUsnRec->RecordLength;
                    pUsnRec =(PUSN_RECORD) (LPBYTE
(pUsnRec)+pUsnRec->RecordLength);
                }
    med.StartFileReferenceNumber=*(USN*)(&buf[0]);
            }
        }
}
```

先用 FSCTL_QUERY_USN_JOURNAL 指令码获取 USN 日志基本信息，存放于 USN_JOURNAL_DATA 结构中。根据其数值在下面的 MFT_ENUM_DATA 结构中设置枚举范围。注意到 MFT 是分多页存储的，因此在读取完一页后，要重设其子项 StartFileReferenceNumber 的内容，这样才能继续下一页操作。为了提高后面检索文件的速度，使用两个散列表做索引。其中 FileGuid 结构的定义是：

```
typedefstruct _FileGuid
{
    UINT64 fid,pid;
    DWORD attr;
}FileGuid,*LPFileGuid;
```

取出每个 USN_RECORD 记录的 FileReferenceNumber 值（文件引用数）放在 fid 中，ParentFileReferenceNumber 值（父目录引用数）放在 pid 中，FileAttributes 值（文件属性）放在 attr 中。

AuxIdx 结构的定义是：

```
typedefstruct _AuxIdx
{
    string name;
    UINT64 pid;
}AuxIdx,*LPAuxIdx;
```

两个结构的 name 存放文件名，未包含全路径，其获取方

法在后面详述。另外 USN_RECORD 的子项 FileName 记录的文件名是宽字符，在使用时需要转换成多字节，为什么没有使用更方便的 W2A，而是直接使用原生 API 函数 WideCharToMultiByte 呢？因为在测试时发现，若是只有数千个文件，使用 W2A 转换耗时还可以忍受，但现在一般人电脑上平均有 30 万个文件，考虑到应用体验，还是使用后者为妙。当然也可考虑直接使用 wstring 记录结果，这样程序执行效能还能略微提升一点。

第 5 步索引全部日志条目并获取全路径。主要思路是使用前述的两个辅助结构，仿照数据库的自然连接方式，利用父子文件引用数，反向递推出全路径，然后存储在数据集合 hash_multimap 中，方便以后查询。在笔记本电脑上测试，索引 24 万个文件，时间是 4 秒左右，速度与 Everything 首次运行时相当，如下所示：

```
voidComposeName(string drv,stringname,vector<NameAttr>
& path,
    hash_multimap<string,vector<NameAttr>>& list,
    hash_multimap<string,FileGuid>&hms,
    hash_multimap<UINT64,AuxIdx>&hmu,
    UINT64 fid0,UINT64 pid0)
{
    auto beg1(hms.lower_bound(name));
    auto end1(hms.upper_bound(name));
    for(auto pos1=beg1;pos1! =end1;++pos1)
    {
        UINT64 tfid(pos1->second.fid);
        UINT64 tpid(pos1->second.pid);
        if(tfid==fid0&&tpid==pid0)
        {
            beg1=pos1;
            break;
        }
    }
    if (0x5000000000005 ==beg1 ->second.pid&&! path.
empty())
    {
        DWORD attr(FILE_ATTRIBUTE_ARCHIVE);
        if(path[0].attr! =0x26&&path[0].attr&attr)
        {
            NameAttrnatr={drv,0};
            path.push_back(natr);
            list.insert(make_pair(path[0].name,path));
            path.pop_back();
        }
}
else
{
    if(beg1! =end1)
    {
        UINT64 pid1(beg1->second.pid);
        auto beg2(hmu.lower_bound(pid1));
        auto end2(hmu.upper_bound(pid1));
        if(beg2! =end2)
        {
            UINT64 pid2(beg2->second.pid);
            UINT64 fid2(beg2->first);
            if(pid1==fid2)
            {
                string file(beg2->second.name);
                DWORD attr(beg1->second.attr);
                NameAttrnatr={file,attr};
                path.push_back(natr);
ComposeName(drv,file,path,list,hms,hmu,fid2,pid2);
                path.pop_back();
            }
        }
    }
}
voidCollectPath (string drv,hash_multimap <string,vector <
NameAttr>>& list,
    hash_multimap<string,FileGuid>&hms,hash_multimap<
UINT64,AuxIdx>&hmu)
{
    vector<NameAttr> path(0);
    for(autopos=hms.begin();pos! =hms.end();++pos)
    {
        DWORD attr(pos->second.attr);
        string name(pos->first);
        NameAttrnatr={name,attr};
        path.push_back(natr);
        UINT64 fid(pos->second.fid);
        UINT64 pid(pos->second.pid);
    ComposeName(drv,name,path,list,hms,hmu,fid,pid);
        vector<NameAttr>(0).swap(path);
    }
}
```

其中函数 ComposeName 是关键所在，CollectPath 函数只是对它的外围调用。注意到 0x5000000000005 是每个磁盘分区的根植，32 位和 64 位 Win7 系统都一样。考虑到 hash_multimap 的实现特点，每次散列同一键值时，得到的内容不一定一样，这就是为什么在 ComposeName 函数开始处，使用 for 循环，直到找出与前次调用时，内容相同的子项为止，这样才能正确应对子目录名与父目录名相同的情况。另外如果想得到与 Everything 一致的结果，可以去掉对 0x26 文件属性的判断。由于每个磁盘分区的 USN 数据都是独立的，因此当本机存在多个分区时，需要多次调用这些函数来分别获取所有日志记录集合。

准备工作都完成后就可以进行实际查询。查询就很简单了，若是查找的内容是完整的文件名，直接散列到

hash_multimap 中即可。若是模糊匹配查询，比如：新建 *.*?t 这个样子，可以先将其改造成正则表达式，再用 regex_match 匹配即可，具体可参考所附例程，不再赘述。另外为了提高代码执行效率，在使用 STL 的 copy 算法时，设计了专用的迭代器。有一点需要注意，在本例程中，必须在写好自己的 operator=重载函数后，再提供一个默认的 operator=重载。若忽略此事，在 Debug 模式下编译运行还是正确的，但在 Release 模式下编译，就会出现如下错误提示：

error C2679: 二进制 "=": 没有找到接受 XXX 类型的右操作数的运算符

error C2582: "operator ="函数在 XXX 中不可用

当然在所提供的默认重载函数中，也不一定非要有所作为，还可以写成这样：void operator= (constex_insert<T1,T2>&) {} 之所以会这样，是因为在两个不同类型的容器之间传输数据，导致两个模板参数类型不同。

作为简单演示，未整理成类，大家可以根据需要修改。所附代码在 VS.NET2010+Win7 64 位下调试通过，不需安装其他 SDK。

（作者：申晓）

（上接第 80 页）

```
            wavfile.SeekToBegin();
                        wavfile.Write (&wavheader,sizeof
(WAVE_HEADER));  //更新 WAVE 文件头
            wavfile.Close();
            bTransform=FALSE;
            m_BUTTON2.SetWindowText("转换");
            m_BUTTON1.EnableWindow(TRUE);
            m_BUTTON3.EnableWindow(TRUE);
        str1.Format("转换结束 \r\n 文 件 名：%s\r\n 总 块 数：%
d\r\n 文 件 大 小 ：% d", (LPCTSTR)fn2,nTotalBlocksRetrieved,
wavheader.nRIFFBytes+8);
            AfxMessageBox(str1);
        }
}
```

### 4.3 播放 APE 文件

```
//播放
void CApe2wavDlg::OnButton3()
{
    if(bPlay)
        bPlay=FALSE;
    else
    {
        bPlay=TRUE;
        m_BUTTON3.SetWindowText("停止");
        m_BUTTON1.EnableWindow(FALSE);
        m_BUTTON2.EnableWindow(FALSE);
        int nBlockAlign = c_APEDecompress_GetInfo
(pAPE, APE_INFO_BLOCK_ALIGN);//块对齐字节数
        int nTotalBlocks = c_APEDecompress_GetInfo
(pAPE, APE_DECOMPRESS_TOTAL_BLOCKS);//需要解码的
//总块数
        int nBlocksRetrieved = 1;
        int nTotalBlocksRetrieved = 0;
        int nRetVal=0;
        unsigned char * pBuffer = new unsigned char
[1024 * nBlockAlign];//解码后数据缓冲区
        MSG msg;
m_SLIDER1.SetRange(0,nTotalBlocks);//设置滑动条
    c_APEDecompress_Seek(pAPE,0);   //定位数据块
    while (nBlocksRetrieved > 0)
    {
        if(PeekMessage(&msg,0,0,0,PM_REMOVE))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
        if(bPlay==FALSE)
            break;
nRetVal = c_APEDecompress_GetData (pAPE, (char*)
pBuffer, 1024, &nBlocksRetrieved);//解码数据块
            if (nRetVal ! = 0)
            {
                AfxMessageBox("播放失败");
                break;
            }
wavout.Play(pBuffer,nBlocksRetrieved*nBlockAlign);//播放
            nTotalBlocksRetrieved += nBlocksRetrieved;
            m_SLIDER1.SetPos(nTotalBlocksRetrieved);
//更新滑动条
        }
        delete []pBuffer;
        bPlay=FALSE;
        m_BUTTON3.SetWindowText("播放");
        m_BUTTON1.EnableWindow(TRUE);
        m_BUTTON2.EnableWindow(TRUE);
    }
}
```

## 5 结语

利用 APE 开发库提供的动态链接库 MACDLL 对 APE 格式的音频文件进行解码，实现 APE 文件的播放并转码为 WAV 文件。程序采用 Visual C++ 6.0 开发，在 Windows XP 和 Windows 8 下测试通过。

（收稿日期：2013-02-26）

# 电脑系统维护经验与技巧

**购买 SSD 后，空盘时该做什么工作**

首先在主板 BIOS 中要开启 AHCI 模式全称 Advanced Host Controller Interface,即高级主机控制器接口，相比老旧的“IDE 虚拟模式”更适合 SSD SATA 存储设备通信协议。

假如主板 BIOS 不开启 AHCI，那么 SATA 设备其实是工作在 IDE 虚拟模式下的，南桥会把 SATA 设备认为是老旧的 IDE 设备，占用主板 IRQ。而通过 AHCI 可激活设备的高级 SATA 功能如原生指令队列（NCQ）和热插拔等。开启 AHCI 模式可以使多线程、并发队列的读写请求。不开启 AHCI 模式下运行 SSD 对其性能影响巨大。

**怎么检查您是否开启了 AHCI 模式**

操作如下：

（1）点击 Win 键+R，进入运行对话框。

（2）输入 Regedit 进入注册表。

（3）选择路径“HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\msahci”。

（4）把 start 值改为 0，默认是 3，直接重启电脑，在 BIOS 中开启 AHCI 模式，正常进入系统。

**固态硬盘安装系统时为什么要“4KB 对齐”**

随着硬盘容量不断扩展，为了提高容错能力和读写速度，将原来的每个扇区 512 字节改为每个扇区 4096 个字节，也就是现在常说的“4KB 扇区”，那“4KB 对齐”就是符合“4KB 扇区”定义格式化过的硬盘，并且按照 4KB 的规则写入数据。在固态硬盘中，如果 4KB 不对齐，写入的数据写入点正好会介于在两个 4KB 扇区的之间，也就是说即使是写入最小量的数据，也会使用到两个 4KB 扇区，这样会造成跨区读写，读写次数放大，从而影响读写速度，另外由于固态硬盘写入次数有限，如果 4KB 不对齐，寿命也会缩短。

**如何进行“4KB 对齐”**

如果用户的固态硬盘是拿来当系统盘使用的，安装系统时可以使用 Win7 磁盘管理，所分得区都是对齐的；用户也可以使用“DiskGenius”软件进行手动指定扇区位置，分区时选择“对齐到下列扇区的整数倍”扇区数选择 8 或者以上，建议选择 2048，这就是 Win7 分区使用的值，实际对齐大小就是 1024K。

**如何检查 SSD 是否已经 4K 对齐**

最简单的方法是看 AS SSD Benchmark 软件分界面。界面上的绿色部分显示 1034240K，其中 103424 数字大小跟分区有关，大小不一，则表示对齐；否则显示红色 bad 字样。

**怎样确定 Trim 指令打开**

Trim 指令也叫 disable delete notify (禁用删除通知)，是微软联合各大 SSD 厂商所开发的一项技术，属于 ata8-acs2 规范的技术指令。Windows 7 系统对支持 Trim 指令的 SSD 启动 Trim 命令后，这个过程变得简单许多。这时在对 SSD 进行数据删除的操作，NTFS 文件系统是不向 SSD 发送删除指令的，固态硬盘发出新的 Trim 操作指令，告诉它相关页面可以安全擦除，得到这一指示后，就不会急于执行擦除操作，而是等到再次有写入操作的时候再执行，因为这时相关区域已经可以重新使用了，就不用花时间去擦除原本的数据。其速度比直接读写硬盘去标记删除区域要快得多，既提升了读写效率又大大减轻了固态硬盘的损耗。

**怎样判断电脑里是否已开启 Trim 指令**

进行以下的操作：

1）点击开始菜单，在搜索栏中输入 CMD；

2）在搜索结果的图标上右键单击，选择以管理员权限运行；

3）输入“fsutil behavior query DisableDeleteNotify”；

4）如果返回值是 0，则代表电脑的 Trim 处于开启状态；

5）如果返回值是 1，则代表电脑的 Trim 处于关闭状态。

**SSD 可完全替代机械硬盘吗**

当 SSD 的价格逐步走低，广大用户对于 SSD 的热情终于被点燃了，现在 120GB 的 SSD 价格都已经降到了 600 元左右。而 SSD 高速的读写性能和良好的抗震性能又为它赢得了很多用户的青睐，因此可以看到目前市场上越来越多的用户开始选择 SSD 来装机。然而 SSD 真的能够完全替代机械硬盘吗？至少在容量方面是不大可能的，毕竟现在 600 多元还是可以买到 1TB 的机械硬盘，在去年同期，600 元甚至可以买到 2TB 的机械硬盘。

从测试结果可以看到，机械硬盘、双机械硬盘 RAID 0 系统以及 SSD 在磁盘性能上的差异是巨大的，单个机械硬盘的磁盘性能确实比较差，但胜在容量巨大。双机械硬盘 RAID 0 系统的持续读写能力相较于单机械梗盘有了成倍的增长，加载游戏程序的速度也有明显的提高，而且容量并没有损失。SSD 几乎所有的性能都超过了两块 7200RPM 硬盘的 RAID 0，而且这还是一款性能相对较低的 SSD，现在的高速 SSD 磁盘性能更好。

## MAIL TO THE DOCTOR

另外，也要看到，除了读写能力以外，对于视频编码和图像处理来说，更换 SSD 或者 RAID 0 系统并不能从根本上解决性能的瓶颈，因为对于这些操作来说处理器性能才是最大的瓶颈。也就是说，有经济实力的读者朋友，当然可以购买大容量的 SSD 来获得最好的机械性能和磁盘性能，动手能力强的读者可以选择多块硬盘组成 RAID 0 系统，而普通读者当然选择普通的机械硬盘最为合适。从应用来说，专注于视频编码和图像处理的用户并不需要读写速度超快的 SSD，相反，对于他们来说容量够大，数据够安全才是最主要的。目前，SSD 并不是在任何应用环境下都能完全替代机械硬盘。

**？ 固态硬盘购买多大容量的为宜**

！ 由于固态硬盘的内部结构所致，容量越小的产品，读写速度就会越慢，容量越大的产品，读写速度则会越快，以金士顿官方 V100 的数据为例，30GB 的最高读写速度仅为 160MB/s 和 70MB/s，64GB 的最高读写速度由达到了 250MB/s 和 145MB/s，128GB 的最高读写速度更是高达 250MB/s 和 230MB/s。所以说，要想体验真正的高速，还是大容量固态硬盘更为适合。不过就目前来看，128GB 固态硬盘通常还要卖到“天价”，所以追求性能，暂时还是购买 64GB 固态硬盘为宜。

**？ SLC 固态硬盘和 MLC 固态硬盘有何区别**

！ 由于固态硬盘刚进入普及阶段，出于价格原因，很多用户选择的还是容量相对较小，性能也相对低一些的 64GB 固态硬盘，不过那还是针对采用 MLC 闪存颗粒的产品而言的。对于采用 SLC 闪存颗粒的固态硬盘来说，又贵又快才是它的特点。

SLC 是指一个 Block（数据块，闪存的基本存储单元）只有两种电荷状态分别表示 0 或者 1，因为只需要一组高低电压就可以区分出 0 或者 1 信号，所以 SLC 最大的驱动电压可以做到很低，结构简单，速度表现更好，而且寿命更长。而 MLC 采用较高的电压驱动，所以不同级别的电压在一个 Block 中记录两组位信息（00、01、11、10），大幅提升了记录密度，但同时一个 Block 存储两组位数据也就需要更长的时间来读写，再考虑到电压控制、CRC 写入方式等因素，因此 MLC 的读写速度也比 SLC 低得多，加上电压变化更频繁，所以寿命也比 SLC 短很多。这也就是说，采用 SLC 闪存颗粒的固态硬盘性能出色但价格要贵两三倍的原因。高性能首选 SLC 固态硬盘。

# 基于 DirectUI 架构的中海油软件平台的用户体验设计与实现

中海油是中国海洋石油总公司的简称，是 1982 年成立的国家石油公司。经过几十年的不断发展，中海油拥有上百个油田工作系统，而每个油田系统均配有功能强大的软件系统作为它的支撑。每套软件系统的不断升级和维护让它们的程序变得更加复杂。由于这些软件都是从各种作业实践中逐步提炼而成的，从一开始就缺少统一的规划和全局的用户体验考虑，所以导致了一套系统一个样。给油田作业的工程师们带来了极大的操作难度。目前，中海油的软件系统的用户体验方面的问题已经严重影响到了海上作业的工作效率。中海油集团的上上下下也对用户体验问题高度关注。

UIPower 在充分了解中海油的后台系统的需求后，建议中海油采用 UI 设计+UI 开发的整体 UI 解决方案。

UI 设计阶段：进行了用户研究、交互设计、用户体验测试和视觉设计等方面的工作。首先解决了软件难用的问题，对软件中的各种窗体进行了重新规划，详细分析了各种控件在实际使用中的频率。接着提出中海油作业软件的整体交互设计规范。从规范上确保未来的各个软件系统保持一致的操作习惯。通过对新的交互设计严格的实验室测试来检验它的科学合理性，去除里面不合理的地方，不断迭代最终形成易上手、易操作的软件交互模型。UIPower 的视觉设计师根据多年的经验对软件进行整体风格、色调、质感等的创意，最终形成在视觉上具有高度专业性的软件高保真设计稿。

UI 开发阶段：对于 UI 改造项目来说，设计出漂亮的界面其实已经很困难了，而要将专业的 UI 设计真正实现出来则是难上加难。原因是客户端各个开发工具对 UI 的实现均支持不足，无法满足各种新特效新质感的窗口或控件的交互和视觉呈现。一般来说，要实现高保真的效果图程序需要花费半年甚至更多的人月数。然而，现实却无法允许这样的界面开发进度。UIPower 一直采用自主研发的 DirectUI 产品作为他们的界面开发的基石。DirectUI 可以将一个软件的所有界面在 1 周内全部实现出来，而且对界面的效果做到没有损失，UIPower 对项目验收的标准是“像素级比对”，即在最后的成型程序与高保真的视觉设计稿通过屏幕的逐像数比对，只要有一对像素的值不一样即认为是不通过验收。正是基于这样的验收标准，UIPower 在业内迅速成为国内大型企业争相合作的对象。

经过 2 个多月的整体改造，中海油集团内的几款重要的软件系统，均已成功上线，并正式投入了使用。从集团上上下下的相关人员对本次的 UI 改造成果表示了高度的认可和极大的满意。

UIPower 于 2004 年成立至今，一直以“让天下没有难做的界面“为使命，先后服务于华为、中兴、盛大网络、中国移动、中国电信、铁道研究院等国内大型企业。UIPower 的阙总在接受采访时说：“未来 UIPower 一直致力于界面产品的研发与升级，全面助力中国软件相关企事业单位，提高中国软件产品在国际市场上的竞争力，实现软件强国的梦想!”。


书名：WebGL 高级编程——开发 Web 3D 图形
ISBN：978-7-302-32183-5
定价：49.00 元
作者：（美）阿尤鲁（Anyuru, A.）著

本书介绍了如何开发基于 WebGL 的 Web 应用程序、以及如何开发并帮助读者快速使用流行本地应用所具有的高品质、实时 3D 图形功能来开发 Web 游戏和应用。本书提供了创建可以在大多数 Web 浏览器上运行的图形丰富的游戏和 Web 应用所需掌握的知识和技能。本书包括清晰、循序渐进的指导，专家提示和技能培养练习，详细的补充信息，实践开发实例等所有基础知识。

WebGL 以前所未有的方式允许用户硬件加速 2D 和 3D 图形。用户既可以应用 HTML、CSS 和 JavaScript 提供的全部优点，同时又可以应用功能强大的图形处理单元（Graphics Processor Unit，GPU）提供的好处。本书向用户介绍开始开发基于 WebGL 的 2D 或 3D 图形应用程序所需的全部基础知识和准备工作。


书名：SQL Server 2012 数据库设计与开发实务
ISBN：978-7-302-31898-9
定价：58.00 元
作者：陈会安 著

本书从数据库系统设计与开发者角度出发，详细深入地介绍了 SQL Server 数据库程序设计与开发的方法和技巧，内容包括数据库系统相关理论、数据库设计理论、T-SQL 语言的语法、预存程序、过程对象、自定义函数、触发程序、数据指针和交易处理。本书完美结合数据库理论与设计实践，除了使用大量图形来介绍数据库系统理论、实体关系模型和正规化外，更以实例介绍数据库设计。读者不仅可以使用数据库设计工具组绘制专业的实体关系图，还可以将设计结果建成 SQL Server 数据库，来验证实体关系模型的数据库设计理论。

本书适用于微软 SQL Server 2012 企业版、标准版以及 Express 版，是数据库设计与开发人员或学校数据库设计相关课程所规划的实用教材和实训指南。


书名：RHCSA/RHCE Red Hat Linux 认证学习指南（第 6 版）：EX200 & EX300
ISBN：978-7-302-31712-8
定价：98 元
作者：（美）江（Jang, M.）著

本书是亚马逊畅销榜领先图书，经过教学检验的 IT 培训和考试准备工具，100%覆盖——300 多道考试真题，覆盖所有 RHCSA 和 RHCE 考试。

本书不仅包含数百道复习题、完全覆盖基于性能的所有要求，而且包含你想知道的一切——为您详细介绍如何准备这两个极具挑战性的考试。100%完全覆盖 EX200 和 EX300 的所有正式目标考试准备检查清单——如何完成该清单上的所有目标，考试就没问题每章的考试部分突出显示主要考试主题用于快速复习的小练习 100 多个实验问题——两套完整的基于实验的 RHCSA 考试题和两套完整的基于实验的 RHCE 考试题——与真实考试的格式、风格、主题和难度相一致。


书名：Android 4.X 应用与开发实战手册（第 2 版）
ISBN：978-7-302-32213-9
定价：58.00 元
作者：黄彬华 著

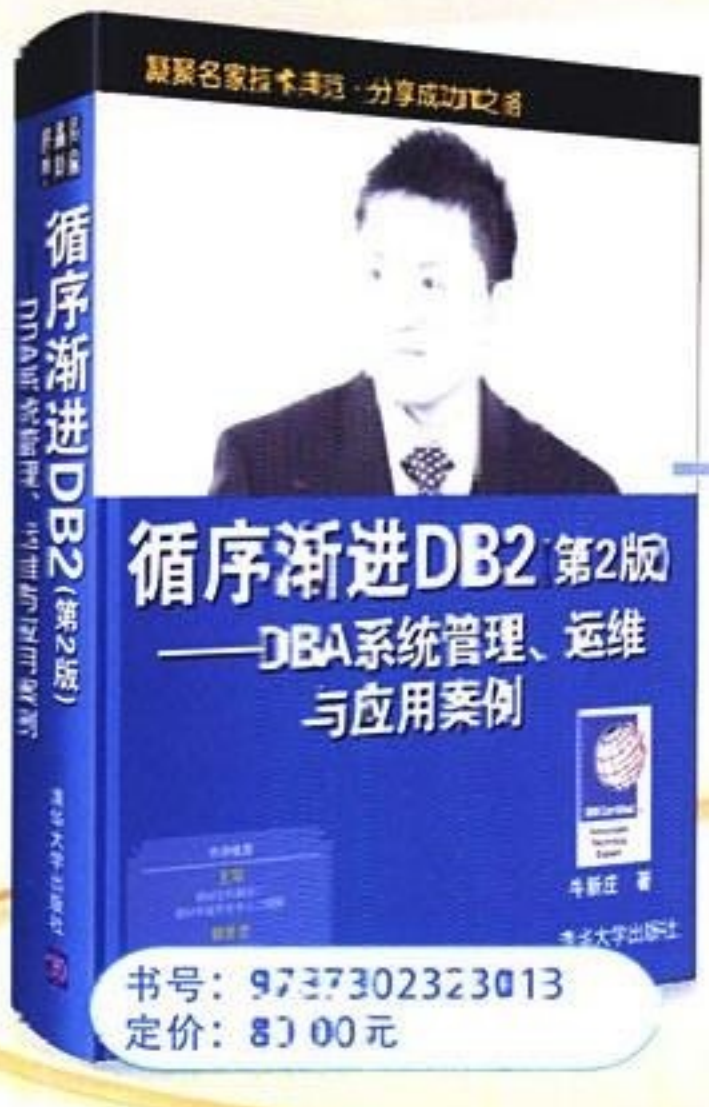
本书以 Android 4.X 进行开发示范，通过大量图示与 step by step 方式，详细介绍了使用 Android 开发智能手机、平板电脑应用程序的方法，读者无须强记即可灵活运用各项开发技巧。本书还介绍了如何将应用程序上传到 Google Play（原 Android Market）供全球 Android 移动设备用户下载，以及如何将 AdMob 广告板置入应用程序，即使应用程序免费也可以获利。此外，本书所有范例都已录制成长达 450 分钟的视频，是帮助初学者学习和教师教学的必备强化工具。

从IT草根到年薪超200万 技术大牛

清华大学出版社
http://www.tup.com.cn

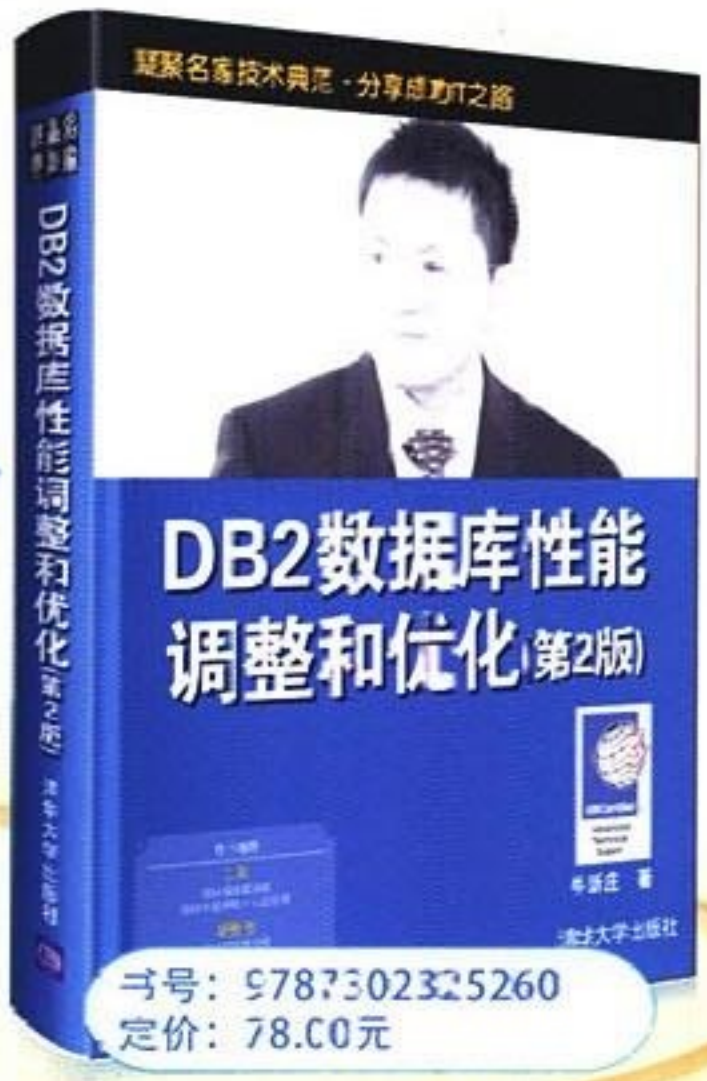
# 国内权威DB2图书"三部曲"

## 业界领军人物 牛新庄博士执笔



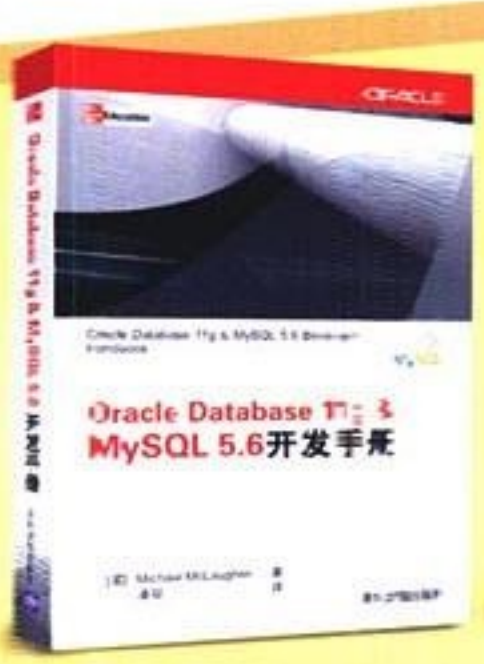

书号：9787302323013
定价：80.00元


书号 9787302323839
定价 68.00元



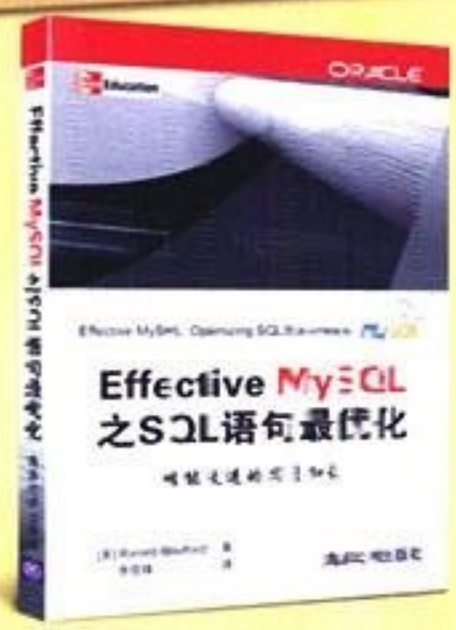

书号：9787302325260
定价：78.00元

## 数据库好书推荐

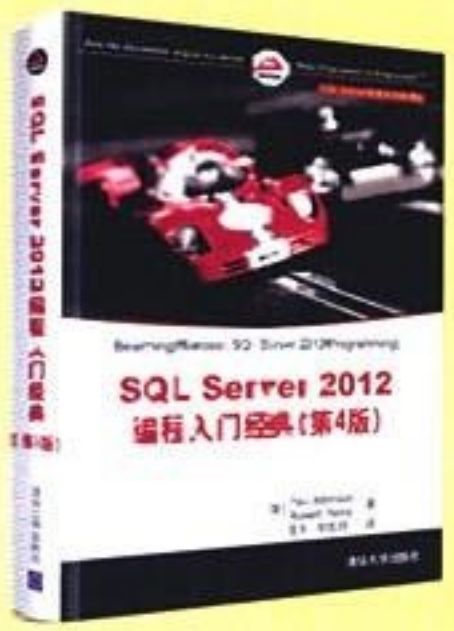



书号：97873023103[illegible]0
定价：79.80元


书号：9787302304258
定价：50.00元



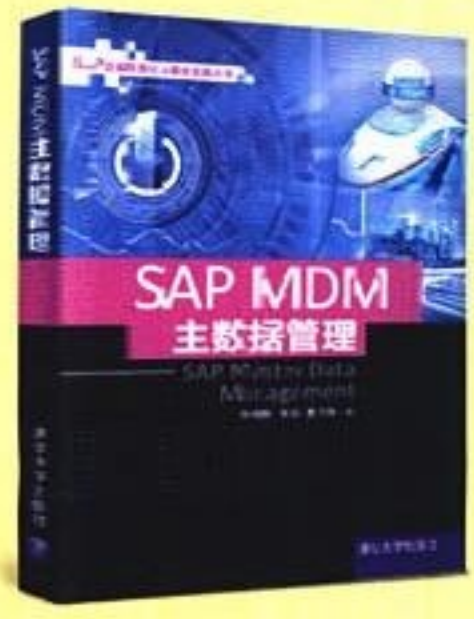

书号：9787302304295
定价：29.00元



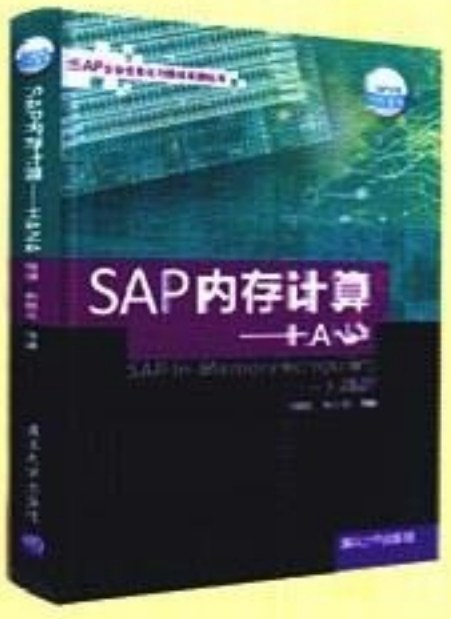

书号：97873023165[illegible]3
定价：89.00元


书号 9787302310143
定价 79.80元


书号：9787302314659
定价：39.00元




书号：9787302317876
定价：98.00元




书号：9787302323570
定价：68.00元


书号：9787302292562
定价：49.80元


书号：9787302311089
定价：79.80元

邮发代号：82-715

# 欢迎订阅 2013 年

# 《电脑编程技巧与维护》半月刊

上半月刊解析主流编程语言典型编程案例，提供编程实践中高手们的经验与技巧。

下半月刊荟萃电脑产学研应用，展现多领域新进展、新方法、新成果。

上、下半月每期均为 11 元

1. 订阅全年（24 期），可享受 8.5 折优惠，原价 264 元，优惠价 225 元。
2. 单独订阅上、下半月（12 期），可享受 9 折优惠，原价 132 元，优惠价 119 元。

官方网址：http://www.comprg.com.cn

## 订阅方式

汇款地址：北京市海淀区长春桥路5号6号楼1209室　收款人：电脑编程技巧与维护杂志社　邮编：100089
电话/传真：32561614　E-mail:zzsfx@vip.sina.com　QQ:565699495
汇款未注明所购买数量和邮寄地址，请与杂志社联系。